# Chapter 5
# Selected multifactorial methods

So far we have only been concerned with monofactorial methods, i.e., methods in which we investigated how maximally one independent variable is correlated with the behavior of one dependent variable. In many cases, proceeding like this is the beginning of the empirical quantitative study of a phenomenon. Nevertheless, such a view on phenomena is usually a simplification: we live in a multifactorial world in which probably no phenomenon is really monofactorial – probably just about everything is influenced by several things at the same time. This is especially true for language, one of the most complex phenomena resulting from human evolution. In this section, we will therefore discuss several multifactorial techniques, which can handle this kind of complexity better than the monofactorial methods discussed so far. You should know, however, each section's method below could easily fill courses for several quarters or semesters, which is why I can impossibly discuss every aspect or technicality of the methods and why I will have to give you a lot of references and recommendations for further study. Also, given the complexity of the methods involved, there will be no discussion of how to compute them manually. Section 5.1, 5.2, and 5.3 introduce multifactorial extensions to the chi-square test of Section 4.1.2.2, correlation and linear regression of Section 3.2.3, and the *t*-test for independent samples of Section 4.3.2.1 respectively. Section 5.4 introduces a method called binary logistic regression, and Section 5.5 introduces an exploratory method, hierarchical agglomerative cluster analysis.

Before we begin to look at the methods in more detail, one comment about multifactorial methods is in order. As the name indicates, you use such methods to explore variation in a multi-variable dataset and this exploration involves formulating a statistical *model* – i.e., a statistical description of the structure in the data – that provides the best possible characterization of the data that does not violate Occam's razor by including more parameters than necessary and/or assuming more complex relations between variables than are necessary. In the examples in Chapter 4, there was little to do in terms of Occam's razor: we usually had only one independent variable with only two levels so we did not have to test whether a simpler approach to the data was in fact better (in the sense of explaining the data just as well but being more parsimonious). In this chapter, the situation will

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

be different. The study of multifactorial approaches typically involves a stepwise procedure where you

- start out from the so-called *maximal model*, i.e., the model that includes all *predictors* (i.e., all variables and their levels and their interactions) that you are considering;
- iteratively delete the least relevant predictors (starting with the highest-order interactions) and fit a new model; until
- you arrive at the so-called *minimal adequate model*, which contains only predictors that are either significant themselves or participate in significant higher-order interactions.

Note that while this principle is in general well-known, it is often not so well recognized for predictors that are neither interactions of variables or variables but variable levels. The above definition of predictors requires that the final, minimal adequate model should not only contain only those *interactions* or *variables* that participate in significant interactions, but also only those *variable levels* that make some significant contribution (recall exercise #3 from the assignments for Chapter 4). In other words, the inclusion of different variable levels should ideally be scrutinized for whether variable levels much be kept apart just as the inclusions of separate variables and interactions should be. Note, though, that a conflation of variable levels must make sense conceptually: it is not useful to create a new combination of variable levels that looks nicer statistically but is conceptually senseless – modeling is a means to an end (namely, understanding the data), not an end in itself. The principle of parsimony is therefore a very important methodological guideline and will surface in different forms in this chapter. I like to think of this so-called model selection approach – but also of statistical approaches in general – as a kind of detective work: you are presented with a mystery (a data set), whose internal structure you need to explore. In order to do that, you look at the data from different angles (using different graphs or different transformations of variables), and just like peeling an onion, you remove layers in the form of unnecessary variables or variable levels until you get to the core.

On the one hand, this makes this kind of approach more difficult than the simple monofactorial tests in Chapter 4, and unfortunately the recommendations on how to proceed differ. Some sources argue in favor of an approach where you rigorously only delete one insignificant parameter at the time, others recommend to proceed not so much on the basis of *p*-values, but on the basis of other statistics (most notably, *AIC*), yet others

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

begin with a maximal model and delete several insignificant parameters at once, and sometimes you find even more than one of these approaches in the same reference. On the other hand, this kind of approach is therefore also much more interesting; there is little more rewarding than, after some long evaluation and model selection process, being able to say, "oh, so this is what's actually going on here …"

**Recommendation(s) for further study:**
Crawley (2007: Ch. 9) is most instructive

## 1. The multifactorial analysis of frequency data

In Section 4.1.2.2, I introduced how you investigate the correlation between nominal or categorical variables, but we only considered two variables at the same time – GIVENNESS and CONSTRUCTION. But of course, you often observe more than one independent variable. There are many different methods that can be used for such data, loglinear analysis, count/Poisson regression, sometimes binary logistic regression, correspondence analysis, association rules, hierarchical configural frequency analysis, and probably others. Since the approach of hierarchical configural frequency analysis is so similar to the already familiar chi-square test, this is the approach to be discussed here, but I already want to recommend to you to read up on count/Poisson regression once you've read Sections 5.2 and 5.4.

### 1.1. Configural frequency analysis

The simple configural frequency analysis (CFA) is an extension of the chi-square test discussed in Section 4.1.2.2. I already mentioned above that the contribution to chi-square is a useful heuristic in the interpretation of the data because the larger the contribution to chi-square of a cell, the more that cell contributes to the overall interaction effect. I also mentioned that it is possible to test contributions to chi-square for significance, and that is the basic idea of a CFA. In some sense, a CFA is both an exploratory and a hypothesis-testing method. It is exploratory because every possible combination of variable levels gets tested for the presence or absence of an effect, and it tests hypotheses because each combination is subjected to a significance test (with a small additional complication to be discussed presently). I

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

will focus here only on cases where there is not necessarily an *a priori* and precisely formulated alternative hypothesis; but the recommendations for further study will point you to readings where such issues are also discussed.

The general procedure of a CFA is this:

| **Procedure** |
|---|
| Tabulating the observed frequencies |
| Computing the contributions to chi-square |
| Computing $p_{corrected}$-values for the contribution to chi-square for $df = 1$ |

We first return to the example from Section 4.1.2.2. You investigated the constituent order alternation of particle placement using data from Peters (2001). You loaded the files from <C:/_sflwr/_inputfiles/04-1-2-2_vpcs.txt>, which are now also in <C:/_sflwr/_inputfiles/05-1-1_vpcs.txt>), you computed a chi-square test, and you inspected the contribution to chi-square:

*Table 41.* 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 |

```
> rm(list=ls(all=T))¶
> VPCs<-read.table(choose.files(),·header=T,·sep="\t"¶
> attach(VPCs)¶

> chisq.test(table(CONSTRUCTION,·REFERENT),·correct=F)¶
·····Pearson's·Chi-squared·test
data:··table(CONSTRUCTION,·REFERENT)
X-squared·=·9.8191,·df·=·1,·p-value·=·0.001727
> chisq.test(table(CONSTRUCTION,·REFERENT),·correct=F)$res^2¶
···········REFERENT
CONSTRUCTION···given·····new
···V_DO_PRt·3.262307··2.846825
···V_PRt_DO·1.981158··1.728841
```

Now, how do you compute the probability not of the chi-square table of the whole table, but of an individual contribution to chi-square? First, you

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

need a *df*-value, which we set to 1. Second, you must correct your significance level for multiple *post-hoc* tests. To explain what that means, we have to briefly go off a tangent and return to Section 1.3.4.

In that section, I explained that the *probability of error* is the probability to obtain the observed result when the null hypothesis is true. This means that probability is also the probability to err in rejecting the null hypothesis. Finally, the *significance level* was defined as the threshold level or probability that the probability of error must not exceed. Now a question: if you reject two independent null hypotheses at each $p = 0.05$, what is the probability that you do so correctly both times?

**THINK**
**BREAK**

This probability is 0.9025, i.e. 90.25% Why? Well, the probability you are right in rejecting the first null hypothesis is 0.95. But the probability that you are always right when you reject the null hypothesis on two independent trials is $0.95^2 = 0.9025$. This is the same logic as if you were asked for the probability to get two sixes when you simultaneously roll two dice: $^1/_6{}^2 = {}^1/_{36}$. If you look at 13 null hypotheses, then the probability that you do not err once if you reject all of them is in fact dangerously close to 0.5, i.e., that of getting heads on a coin toss: $0.95^{13} \approx 0.5133$, which is pretty far away from 0.95. Thus, the probability of error you use to evaluate each of the 13 null hypotheses should better not be 0.05 – it should be much smaller so that when you perform all 13 tests, your overall probability to be always right is 0.95. It is easy to show which probability of error you should use instead of 0.05. If you want to test 13 null hypotheses, you must use $p = 1\text{-}0.95^{(1/13)} \approx 0.00394$. Then, the probability that you are right on any one rejection is 1-0.00394 = 0.99606, and the probability that you are right with all 13 rejections is $0.99606^{13} \approx 0.95$. A shorter heuristic that is just as conservative (some say, too conservative) is the Bonferroni correction. It consists of just dividing the desired significance level – i.e., usually 0.05 – by the number of tests – here 13. You get $^{0.05}/_{13} \approx 0.003846154$, which is close (enough) to the exact probability of 0.00394 computed above. Thus, if you do multiple *post hoc* tests on a dataset, you must adjust the significance level, which makes it harder for you to get significant results just by fishing around in your data. Note, this does not just apply to a (H)CFA – this is a general rule! If you do many *post hoc* tests, this means that the adjustment

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

will make it very difficult for you to still get any significant result at all, which should motivate you to formulate reasonable alternative hypotheses beforehand rather than endlessly perform *post hoc* tests.

Let's return to the data from Peters (2001). Table 41 has four cells which means that the *post hoc p*-value you would need according to the Bonferroni correction is $^{0.05}/_{4} = 0.0125$ (or, if you want to be as exact as possible, $1\text{-}0.95^{(1/4)} \approx 0.01274146$). What is therefore the contribution to chi-square value you need to find in the table (for $df = 1$)? And what are the similarly adjusted chi-square values for $p = 0.01$ and $p = 0.001$?

**THINK**
**BREAK**

```
> qchisq(c(0.0125,·0.0025,·0.00025),·1,·lower.tail=F)·#·or¶
> qchisq(c(0.05,·0.01,·0.001)/4,·1,·lower.tail=F)¶
[1]··6.238533··9.140593·13.412148
```

Now you can check which contribution to chi-square exceeds which of these values. You find that none of them does, which means that the overall interaction is of course still significant but that no single cell reaches a standard level of significance.

```
> which(chisq.test(table(CONSTRUCTION,·REFERENT),·
      correct=F)$res^2>6.239)¶
integer(0)
```

Let us now look at a more complex and thus more interesting example. As you know, you can express relations such as possession in English in several different ways, the following two of which we are interested in.

(53) a. the president's speech (*s*-genitive) = $NP_{Possessor}$'s $NP_{Possessed}$
b. the speech of the president (*of*-construction) = $NP_{Possessed}$ of $NP_{Possessor}$

Since again often both constructions are possible,[33] one might again be

33. These two constructions can of course express many different relations, not just those of possessor and possessed. Since these two are very basic and probably the archetypal relations of these two constructions, I use these two labels as convenient cover terms.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

interested in determining how speakers choose which construction to use. Previous studies have shown that, among other things, the degree of animacy/concreteness of the referents of the possessor and the possessed influences the constructional choice. To evaluate such results, let's assume you extracted 150 examples of each construction from a corpus and coded them with regard to the construction, but also with regard to whether the possessor is an animate being, a concrete object, or an abstract entity (POSSESSOR: *ANIMATE* vs. *CONCRETE* vs. *ABSTRACT*), and the same for the possessed (POSSESSED: *ANIMATE* vs. *CONCRETE* vs. *ABSTRACT*). Table 42 crosstabulates the observed frequencies in the form of a *k* (rows) × *m* (columns) × *s* (slices) table.

*Table 42*. POSSESSOR × POSSESSED × GENITIVE

| *POSSESSED* | *ABSTRACT* | | *CONCRETE* | | *ANIMATE* | | Totals | | |
|---|---|---|---|---|---|---|---|---|---|
| *POSSESSOR* | *OF* | *S* | *OF* | *S* | *OF* | *S* | *OF* | *S* | Totals |
| *ABSTRACT* | 80 | 37 | 9 | 8 | 3 | 2 | 92 | 47 | 139 |
| *CONCRETE* | 22 | 0 | 20 | 1 | 0 | 0 | 42 | 1 | 43 |
| *ANIMATE* | 9 | 58 | 1 | 35 | 6 | 9 | 16 | 102 | 118 |
| Totals | 111 | 95 | 30 | 44 | 9 | 11 | 150 | 150 | 300 |
| | 206 | | 74 | | 20 | | | | |

A CFA now tests whether the observed frequencies of the so-called *configurations* – variable level combinations – are larger or smaller than expected by chance. If a configuration is more frequent than expected, it is referred to as a *type*; if it is less frequent than expected, it is referred to as an *antitype*. In this example, this means you test which configurations of a construction with a particular possessor and a particular possessed are preferred and which are dispreferred.

First, for convenience's sake, the data are transformed into the tabular format of Table 43, whose four left columns contain the same data as Table 42. The column "expected frequency" contains the expected frequencies, which were computed in exactly the same way as for the two-dimensional chi-square test: you multiply all totals of a particular cell and divide by $n^{number\ of\ variables-1}$. For example, for the configuration POSSESSOR: *ABSTRACT*, POSSESSED: *ABSTRACT*, GENITIVE: *OF* you multiply 139 (the overall frequency of abstract possessors) with 206 (the overall frequency of abstract possesseds) with 150 (the overall frequency of *of*-constructions), and then you divide that by $300^{(3-1)}$, etc. The rightmost column contains the contributions to chi-square, which add up to the chi-square value of 181.47 for the complete table.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

*Table 43*. CFA: POSSESSOR × POSSESSED × GENITIVE

| Configuration | | | observed frequency | expected frequency | contrib. to chi-square |
|---|---|---|---|---|---|
| POSSESSOR | POSSESSED | GENITIVE | | | |
| *ABSTRACT* | *ABSTRACT* | *OF* | 80 | 47.72 | 21.83 |
| *ABSTRACT* | *ABSTRACT* | *S* | 37 | 47.72 | 2.41 |
| *ABSTRACT* | *CONCRETE* | *OF* | 9 | 17.14 | 3.87 |
| *ABSTRACT* | *CONCRETE* | *S* | 8 | 17.14 | 4.88 |
| *ABSTRACT* | *ANIMATE* | *OF* | 3 | 4.63 | 0.58 |
| *ABSTRACT* | *ANIMATE* | *S* | 2 | 4.63 | 1.5 |
| *CONCRETE* | *ABSTRACT* | *OF* | 22 | 14.76 | 3.55 |
| *CONCRETE* | *ABSTRACT* | *S* | 0 | 14.76 | 14.76 |
| *CONCRETE* | *CONCRETE* | *OF* | 20 | 5.3 | 40.73 |
| *CONCRETE* | *CONCRETE* | *S* | 1 | 5.3 | 3.49 |
| *CONCRETE* | *ANIMATE* | *OF* | 0 | 1.43 | 1.43 |
| *CONCRETE* | *ANIMATE* | *S* | 0 | 1.43 | 1.43 |
| *ANIMATE* | *ABSTRACT* | *OF* | 9 | 40.51 | 24.51 |
| *ANIMATE* | *ABSTRACT* | *S* | 58 | 40.51 | 7.55 |
| *ANIMATE* | *CONCRETE* | *OF* | 1 | 14.55 | 12.62 |
| *ANIMATE* | *CONCRETE* | *S* | 35 | 14.55 | 28.73 |
| *ANIMATE* | *ANIMATE* | *OF* | 6 | 3.93 | 1.09 |
| *ANIMATE* | *ANIMATE* | *S* | 9 | 3.93 | 6.53 |
| Totals | | | 300 | 300 | 181.47 |

For $df = (k \cdot m \cdot s)-(k+m+s)+number\ of\ variables-1 = 12$, this chi-square value is highly significant:

```
> pchisq(181.47,·12,·lower.tail=F)¶
[1]·2.129666e-32
```

This means the global null hypothesis can be rejected: there is a significant interaction between the animacy/concreteness of the possessor, of the possessed, and the choice of construction ($\chi^2 = 181.47$; $df = 12$; $p < 0.001$). But how do you now decide whether a particular contribution to chi-square is significant and, therefore, indicative of a potentially interesting type or antitype? You compute the adjusted significance levels, from those you compute the adjusted critical chi-square values that need to be exceeded for significant types and antitypes, and then you check which of the contributions to chi-square exceed these adjusted critical chi-square values:

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> qchisq(c(0.05,·0.01,·0.001)/18,·1,·lower.tail=F)¶
[1]··8.947972·11.919293·16.248432
```

- types:
  - POSSESSED: *ABSTRACT of* POSSESSOR: *ABSTRACT* (***);
  - POSSESSED: *CONCRETE of* POSSESSOR: *CONCRETE* (***);
  - POSSESSOR: *ANIMATE 's* POSSESSED: *CONCRETE* (***);
- antitypes:
  - POSSESSOR: *CONCRETE 's* POSSESSED: *ABSTRACT* (**);
  - POSSESSED: *CONCRETE of* POSSESSOR: *ANIMATE* (**);
  - POSSESSED: *ABSTRACT of* POSSESSOR: *ANIMATE* (***).

Thus, in addition to the rejection of the global null hypothesis, there are significant types and antitypes: animate entities are preferred as possessors of concrete entities in *s*-genitives whereas abstract and concrete possessors prefer *of*-constructions. More comprehensive analysis can reveal more, and we will revisit this example shortly. There is no principled upper limit to the number of variables and variable levels CFAs can handle (as long as your sample is large enough, and there are also extensions to CFAs that allow for slightly smaller samples) so this method can be used to study very complex patterns, which are often not taken into consideration.

Before we refine and extend the above analysis, let us briefly look at how such data can be tabulated easily. First, load the data from <C:/_sflwr/_inputfiles/05-1-1_genitives.txt>:

```
> rm(list=ls(all=T))¶
> Genitive<-read.table(choose.files(),·header=T,·sep="\t",·
      comment.char="")¶
> attach(Genitive)¶
> str(Genitive)¶
'data.frame':···300·obs.·of··4·variables:
·$·CASE·····:·int··1·2·3·4·5·6·7·8·9·10·...
·$·POSSESSOR:·Factor·w/·3·levels·"abstract","animate",..:·...
·$·POSSESSED:·Factor·w/·3·levels·"abstract","animate",..:·...
·$·GENITIVE·:·Factor·w/·2·levels·"of","s":·1·1·1·1·1·1·1·...
```

The simplest ways to tabulate the data involves the functions `table` and `prop.table`, which you already know. For example, you can use `table` with more than two variables. Note how the order of the variable names influences the structure of the tables that are returned.

```
> table(GENITIVE,·POSSESSOR,·POSSESSED)¶
> table(POSSESSOR,·POSSESSED,·GENITIVE)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

Note also what this gives you, it will be important later:

```
> table(POSSESSOR, POSSESSED, GENITIVE)[,,1]¶
> table(POSSESSOR, POSSESSED, GENITIVE)[,,"of"]¶
```

The function `ftable` offers another interesting way to tabulate. For our present purposes, this function takes three kinds of arguments:

- it can take a data frame and then cross-tabulates all variables so that the levels of the left-most variable vary the slowest;
- it can take several variables as arguments and then cross-tabulates them such that the levels of the left-most variable vary the slowest;
- it can take a formula in which the dependent variable and independent variable(s) are again on the left and the right of the tilde respectively.

```
> ftable(POSSESSOR, POSSESSED, GENITIVE)¶
> ftable(GENITIVE~POSSESSOR+POSSESSED)¶
                  GENITIVE of  s
POSSESSOR POSSESSED
abstract  abstract          80 37
          animate            3  2
          concrete           9  8
animate   abstract           9 58
          animate            6  9
          concrete           1 35
concrete  abstract          22  0
          animate            0  0
          concrete          20  1
```

You can combine this approach with `prop.table`. In this case, it would be useful to be able to have the row percentages (because these then show the proportions of the constructions):

```
> prop.table(ftable(GENITIVE~POSSESSOR+POSSESSED), 1)¶
                  GENITIVE        of         s
POSSESSOR POSSESSED
abstract  abstract          0.68376068 0.31623932
          animate           0.60000000 0.40000000
          concrete          0.52941176 0.47058824
animate   abstract          0.13432836 0.86567164
          animate           0.40000000 0.60000000
          concrete          0.02777778 0.97222222
concrete  abstract          1.00000000 0.00000000
          animate                 NaN        NaN
          concrete          0.95238095 0.04761905
```

Many observations we will talk about below fall out from this already.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

You can immediately see that, for POSSESSOR: *ABSTRACT* and POSSESSOR: *CONCRETE*, the percentages of GENITIVE: *OF* are uniformly higher than those of GENITIVE: *S*, while the opposite is true for POSSESSOR: *ANIMATE*.

### 1.2. Hierarchical configural frequency analysis

The kind of approach discussed in the last section is a method that can applied to high-dimensional interactions. However, the alert reader may have noticed two problematic aspects of it. First, we looked at all configurations of the three variables – but we never determined whether an analysis of two-way interactions would actually have been sufficient; recall Occam's razor and the comments regarding model selection from above. Maybe it would have been enough to only look at POSSESSOR × GENITIVE because this interaction would have accounted for the constructional choices sufficiently. Thus, a more comprehensive approach would test:

- the three-way interaction POSSESSOR × POSSESSED × GENITIVE (which is what we did);
- a two-way interaction POSSESSOR × POSSESSED;
- a two-way interaction POSSESSOR × GENITIVE;
- a two-way interaction POSSESSED × GENITIVE.[34, 35]

Second, the larger the numbers of variables and variable levels, the larger the required sample size since (i) the chi-square approach requires that most expected frequencies are greater than or equal to 5 and (ii) with small samples, significant results are hard to come by.

Many of these issues can be addressed fairly unproblematically. With regard to the former problem, you can of course compute CFAs of the above kind for every possible subtable, but even in this small example this is somewhat tedious. This is why the files from companion website of this book include an interactive script you can use to compute CFAs for all possible interactions, a so-called *hierarchical configural frequency analysis*. Let us apply this method to the genitive data.

34. Since we are interested in the constructional choice, the second of these interactions is of course not particularly relevant.
35. Strictly speaking, you should also test whether all three levels of POSSESSOR and POSSESSED are needed, and, less importantly, you can also look at each variable in isolation.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

Start R, maximize the console, and enter this line (check `?source`):

```
> source("C:/_sflwr/_scripts/05-1_hcfa_3-2.r")¶
```

Then you enter `hcfa()¶`, which starts the function HCFA 3.2. Unlike most other R functions, HCFA 3.2 is interactive so that the user is prompted to enter the information the script requires to perform the analysis. Apart from two exceptions, the computations are analogous to those from Section 5.1.1 above.

After a brief introduction and some acknowledgments, the function explains which kinds of input it accepts. Either the data consist of a raw data list of the familiar kind (without a first column containing case numbers, however!), or they are in the format shown in the four left columns of Table 43; the former should be the norm. Then, the function prompts you for a working directory, and you should enter an existing directory (e.g., <C:/_sflwr/_inputfiles>) and put the raw data file <05-1-2_genitives.txt> in there.

Then, you have to answer several questions. First, you must specify the format of the data. Since the data come in the form of a raw data list, you enter `1¶`. Then, you must specify how you want to adjust the *p*-values for multiple *post hoc* tests. Both options use exact binomial tests of the kind discussed in Sections 1.3.4.1 and 4.3.1.2. The first option is the Bonferroni correction from above, the second is the so-called Holm adjustment, which is just as conservative – i.e., it also guarantees that you do not exceed an overall probability of error of 0.05 – but can detect more significant configurations than the Bonferroni correction. The first option is only included for the sake of completeness, you should basically always use the Holm correction: `2¶`.

As a next step, you must specify how the output is to be sorted. You can choose the effect size measure *Q* (1), the observed frequencies (2), the *p*-values (3), the contributions to chi-square (4), or simply nested tables (5). I recommend option 1 (or sometimes 5): `1¶`. Then, you choose the above input file. R shows you the working directory you defined before, choose the relevant input file. From that table, R determines the number of subtables that can be generated, and in accordance with what we said above, in this case these are 7. R generates files containing these subtables and saves them into the working directory.

Then, you are prompted which of these tables you want to include in the analysis. Strictly speaking, you would only have to include the following:

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

POSSESSOR × GENITIVE, POSSESSED × GENITIVE, POSSESSOR × POSSESSED × GENITIVE. Why?

**THINK BREAK**

Again, the variables in isolation tell you little you don't already know – you already know that there are 150 *s*-genitives and 150 *of*-constructions – or they do not even involve the constructional choice. Second, this is of course also true of the interaction POSSESSOR × POSSESSED. Just for now, you still choose all tables that are numbered from <0001*.txt> to <0007*.txt> in the working directory.

Now you are nearly done: the function does the required computations and finally asks you which of the working tables you want to delete (just for housekeeping). Here you can specify, for example, the seven interim tables since the information they contain is also part of the three output files the script generated. That's it.

Now, what is the output and how can you interpret and summarize it? First, the file <HCFA_output_sum.txt> from the working directory (or the file I prepared earlier, <C:/_sflwr/_outputfiles/05-1-2_genitives_HCFA_output_sum.txt>. This file provides summary statistics for each of the seven subtables, namely the results of a chi-square test (plus a *G*-square test statistic that I am not going to discuss here). Focusing on the three relevant tables, you can immediately see that the interactions POSSESSOR × GENITIVE and POSSESSOR × POSSESSED × GENITIVE are significant, but POSSESSED × GENITIVE is not significant. This suggests that POSSESSED does not seem to play an important role for the constructional choice directly but, if at all, only in the three-way interaction, which you will examine presently. This small overview already provides some potentially useful information.

Let us now turn to <05-1-2_genitives_HCFA_output_complete.txt>. This file contains detailed statistics for each subtable. Each subtable is reported with columns for all variables but the variable(s) not involved in a statistical test simply have periods instead of their levels. Again, we focus on the three relevant tables only.

First, POSSESSOR × GENITIVE (beginning in line 71). You again see that this interaction is highly significant, but you also get to see the exact distribution and its evaluation. The six columns on the left contain information

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

of the kind you know from Table 43. The column "Obs-exp" shows how the observed value compares to the expected one. The column "P.adj.Holm" provides the adjusted *p*-value with an additional indicator in the column "Dec" (for *decision*). The final column labeled "Q" provides the so-called coefficient of pronouncedness, which indicates the size of the effect: the larger *Q*, the stronger the configuration. You can see that, in spite of the correction for multiple *post hoc* tests, all six configurations are at least very significant. The *of*-construction prefers abstract and concrete possessors and disprefers animate possessors while the *s*-genitive prefers animate possessors and disprefers abstract and concrete ones.

Second, POSSESSED × GENITIVE. The table as a whole is insignificant as is each cell: the *p*-values are high, the *Q*-values are low.

Finally, POSSESSOR × POSSESSED × GENITIVE. You already know that this table represents a highly significant interaction. Here you can also see, however, that the Holm correction identifies one significant configuration more than the more conservative Bonferroni correction above. Let us look at the results in more detail. As above, there are two types involving *of*-constructions (POSSESSED: *ABSTRACT of* POSSESSOR: *ABSTRACT* and POSSESSED: *CONCRETE of* POSSESSOR: *CONCRETE*) and two antitypes (POSSESSED: *CONCRETE of* POSSESSOR: *ANIMATE* and POSSESSED: *ABSTRACT of* POSSESSOR: *ANIMATE*). However, the noteworthy point here is that these types and antitypes of the three-way interactions do not tell you much that you don't already know from the two-way interaction POSSESSOR × GENITIVE. You already know from there that the *of*-construction prefers POSSESSOR: *ABSTRACT* and POSSESSOR: *CONCRETE* and disprefers POSSESSOR: *ABSTRACT*. That is, the three-way interaction does not tell you much new about the *of*-construction. What about the *s*-genitive? There are again two types (POSSESSOR: *ANIMATE 's* POSSESSED: *CONCRETE* and POSSESSOR: *ANIMATE 's* POSSESSED: *ABSTRACT*) and one antitype (POSSESSOR: *CONCRETE 's* POSSESSED: *ABSTRACT*). But again, this is not big news: the two-way interaction already revealed that the *s*-genitive is preferred with animate possessors and dispreferred with concrete possessors, and there the effect sizes were even stronger.

Finally, what about the file <05-1-2_genitives_HCFA_output_hierarchical.txt>? This file is organized in a way that you can easily import it into spreadsheet software. As an example, cf. the file <05-1-2_genitives_hierarchical.ods>. In the first sheet, you find all the data from <05-1-2_genitives_HCFA_output_hierarchical.txt> without a header or footer. In the second sheet, all configurations are sorted according to column J, and all types and antitypes are highlighted in blue and red respectively. In the third

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

and final sheet, all non-significant configurations have been removed and all configurations that contain a genitive are highlighted in bold. With this kind of highlighting, even complex data sets can be analyzed relatively straightforwardly.

To sum up: the variable POSSESSED does not have a significant influence on the choice of construction and even in the three-way interaction it provides little information beyond what is already obvious from the two-way interaction POSSESSOR × GENITIVE. This example nicely illustrates how useful a multifactorial analysis can be compared to a simple chi-square test.

**Recommendation(s) for further study**

- Krauth (1993), Lautsch and von Weber (1995) (both in German), and von Eye (2002) on CFAs
- Agresti (2002) for an overview of many different kinds of analysis for categorical data
- the functions `pairs` and `mosaicplot` as well as `assoc` (from the `library(vcd)`) to explore multidimensional frequency tables graphically
- the functions `loglin` and the function `loglm` (from the `library(MASS)`) to compute loglinear analyses
- the functions `ca` and `mjca` (from the `library(ca)`) to compute correspondence analyses

## 2. Multiple regression analysis

In Sections 3.2.3 and 4.4.1, we looked at how to compute and evaluate the correlation between an independent ratio-scaled variable and a dependent ratio-scaled variable using the Pearson product-moment correlation coefficient *r* and linear regression. In this section, we will extend this to the case of multiple independent variables.[36] Our case study is an exploration of how to predict speakers' reaction times to nouns in a lexical decision task and involves the following ratio-scaled variables:[37]

36. In spite of their unquestionable relevance, I can unfortunately not discuss the issues of repeated measures and fixed/random effects in this introductory textbook without raising the overall degree of difficulty considerably. For repeated-measures ANOVAs, Johnson (2008: Section 4.3) provides a very readable introduction; for mixed effects, or multi-level, models, cf. esp. Gelman and Hill (2006), but also Baayen (2008: Ch. 7) and Johnson (2008: Sections 7.3 and 7.4).
37. The words (but not the reaction times) are borrowed from a data set from Baayen's

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

- a dependent variable, namely the reaction time to words in a lexical decision task REACTTIME, whose correlation with the following independent variables you are interested in;
- an independent variable NUMBERLETTERS, which corresponds to the number of letters of each stimulus word;
- an independent variable KF-WRITTENFREQ, which corresponds to their frequency (according to Kučera and Francis 1967);
- an independent variable FAMILIARITY, which is a familiarity index derived from merging three different familiarity norms;
- an independent variable CONCRETENESS, which reflects the rated concreteness from merging three different studies;
- an independent variable IMAGEABILITY, which indicates the imageability of the referent of the word from the same three studies;
- an independent variable MEANINGFULNESS, which indicates the meaningfulness rating of the stimulus word.

This is the overall procedure of a multiple linear regression:

| **Procedure** |
|---|
| Formulating the hypotheses |
| Computing the observed correlations and inspecting graphs |
| Testing the main assumption(s) of the test: |
| the variances of the residuals are homogeneous and normally distributed in the populations from which the samples were taken or, at least, in the samples themselves |
| the residuals are normally distributed (with a mean of 0) in the populations from which the samples were taken or, at least, in the samples themselves |
| Computing the multiple correlation $R^2$ and the regression parameters |
| Computing the test statistic $F$, the degrees of freedom $df$, and the probability of error $p$ |

We begin, as usual, with the hypotheses. Let us assume we wish to test each independent variable's effect in isolation as well as all interactions of maximally two variables. To keep the hypotheses short, we use an exten-

excellent (2008) introduction, and all other characteristics of these words were taken from the MRC Psycholinguistic Database; cf. <http://www.psy.uwa.edu.au/ mrcdatabase/mrc2.html> for more detailed explanations regarding the variables.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

sion of the coefficient of determination $r^2$, namely its multiple regression equivalent multiple $R^2$:

$H_0$: There is no correlation between REACTTIME on the one hand and the independent variables and their pairwise interactions on the other hand: multiple $R^2 = 0$.

$H_1$: There is a correlation between REACTTIME on the one hand and at least one of the independent variables and/or of their pairwise interactions on the other hand: multiple $R^2 > 0$.

You clear the memory (if you did not already start a new instance of R) and load the data from the file <C:/_sflwr/_inputfiles/05-2_reactiontimes.txt>, note how the stimulus words are used as row names:

```
> rm(list=ls(all=T))¶
> ReactTime<-read.table(choose.files(),·header=T,·sep="\t",·
      row.names=1,·comment.char="",·quote="")¶
> summary(ReactTime)¶
··REACTTIME·····NO_LETT·····KF_WRITFREQ·
Min.···:523.0···Min.···:·3.000···Min.···:··0.00
1st·Qu.:589.4···1st·Qu.:·5.000···1st·Qu.:··1.00
Median·:617.7···Median·:·6.000···Median·:···3.00
Mean···:631.9···Mean···:·5.857···Mean···:··8.26
3rd·Qu.:663.6···3rd·Qu.:·7.000···3rd·Qu.:··9.00
Max.···:794.5···Max.···:10.000···Max.···:117.00
·FAMILIARITY·····CONCRETENESS····IMAGEABILITY···MEANINGFUL_CO
Min.···:393.0···Min.···:564.0···Min.···:446.0···Min.···:315.0
1st·Qu.:470.5···1st·Qu.:603.5···1st·Qu.:588.0···1st·Qu.:409.8
Median·:511.0···Median·:613.5···Median·:604.0···Median·:437.5
Mean···:507.4···Mean···:612.4···Mean···:600.5···Mean···:436.0
3rd·Qu.:538.5···3rd·Qu.:622.0···3rd·Qu.:623.0···3rd·Qu.:466.2
Max.···:612.0···Max.···:662.0···Max.···:644.0···Max.···:553.0
NA's···:·22.0···NA's···:·25.0···NA's···:·24.0···NA's···:·29.0
```

For numerical vectors, the function `summary` returns the summary statistics you already saw above; for factors, it returns the frequencies of the factor levels and also provides the number of cases of `NA`, of which there are a few. Before running multifactorial analyses, it often makes sense to spend some more time on exploring the data to avoid falling prey to outliers or other noteworthy datapoints (recall Section 3.2.3). There are several useful ways to explore the data. One involves plotting pairwise scatterplots between columns of a data frame using `pairs` (this time not from the `library(vcd)`). You add the arguments `labels=…` and summarize the overall trend with a smoother (`panel=panel.smooth`) in Figure 59:

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> pairs(ReactTime,·labels=c("Reaction\ntime",·"Number\nof·
      letters",·"Kuc-Francis\nwritten·freq",·"Familiarity",·
      "Concreteness",·"Imageability",·"Meaningfulness"),·
      panel=panel.smooth)¶
```

You immediately get a first feel for the data. For example, FAMILIARITY exhibits a negative trend, and so does IMAGEABILITY. On the other hand, NUMBERLETTERS shows a positive trend, and CONCRETENESS and KF-WRITTENFREQ appear to show no clear patterns. However, since word frequencies are usually skewed and we can see there are some outlier frequencies in the data, it makes sense here to log the frequencies (which linearizes them) and see whether that makes a difference in the correlation plot (we add 1 before logging to take care of zeroes):

```
> ReactTime[,3]<-log(ReactTime[,3]+1)¶
> pairs(ReactTime,·labels=c("Reaction\ntime",·"Number\nof·
      letters",·"Kuc-Francis\nwritten·freq",·"Familiarity",·
      "Concreteness",·"Imageability",·"Meaningfulness"),·
      panel=panel.smooth)¶
```

As you can see (I do not show this second scatterplot matrix here), there is now a correlation between KF-WRITTENFREQ and REACTTIME of the kind we would intuitively expect, namely a negative one: the more frequent the word, the shorter the reaction time (on average). We therefore continue with the logged values.

To quantify the relations, you can also generate a pairwise correlation matrix with the already familiar function `cor`. Since you have missing data here, you can instruct `cor` to disregard the missing data only from each individual correlation. Since the output is rather voluminous, I only show the function call here. You can see some correlations of the independent variables with REACTTIME that look promising …

```
> round(cor(ReactTime,·method="pearson",·
      use="pairwise.complete.obs"),·2)¶
```

It is also obvious, however, that there are some data points which deviate from the main bulk of data points. (Of course, that was already indicated to some degree in the above summary output). For example, there is one very low value of IMAGEABILITY value. It could therefore be worth the effort to also look at how each variable is distributed. You can do that using boxplots, but in a more comprehensive way than before. First, you will use

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

only one line to plot all boxplots, second, you will make use the numerical output of boxplots (which so far I haven't even told you about):

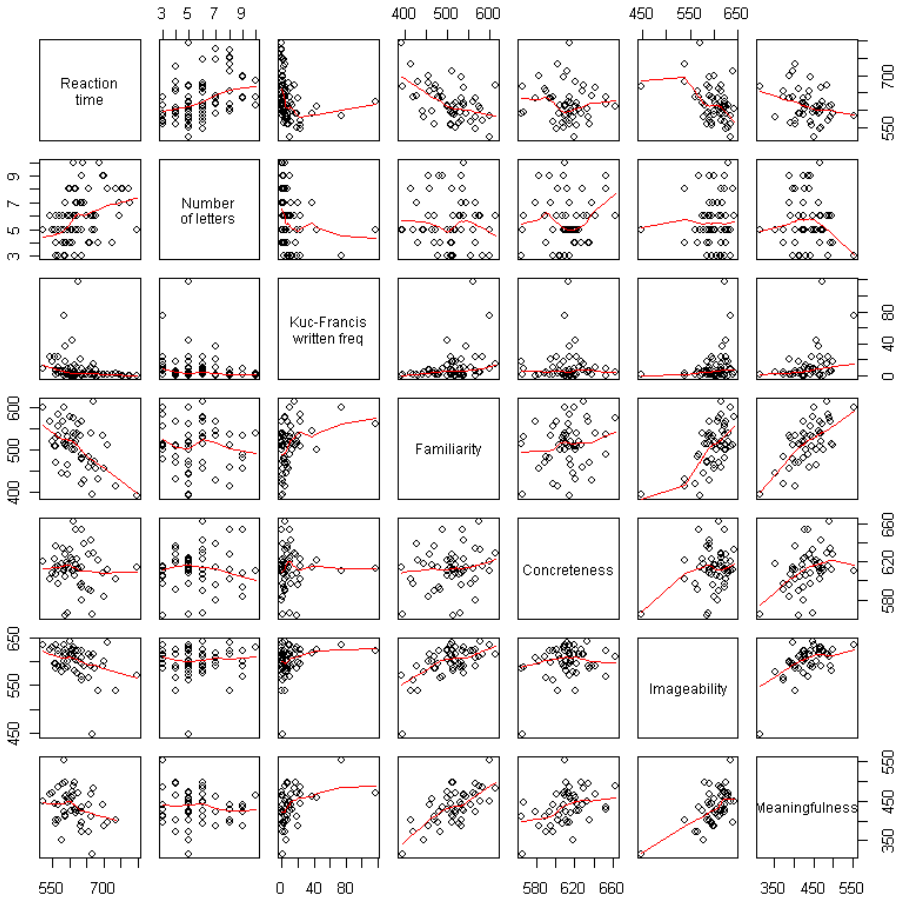


*Figure 59.* Pairwise scatterplots

```
> par(mfrow=c(2,4))¶
> boxplot.output<-apply(ReactTime,·2,·boxplot)¶
> plot(c(0,·2),·c(1,·7),·xlab="",·ylab="",·main="'legend'",·
       type="n",·axes=F);·text(1,·7:1,·labels=paste(1:7,·
       names(ReactTime),·sep="="))¶
> par(mfrow=c(1,·1))·#·restore·the·standard·plotting·setting¶
```

Two things happened here, one visibly, the other invisibly. The visible thing is the graph: seven boxplots were created, each in one panel, and the last of the eight panels contains a legend that tells you for the number of

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

each boxplot which variable it represents (note how `paste` is used to paste the numbers from one to seven together with the column names, separated by a "=").

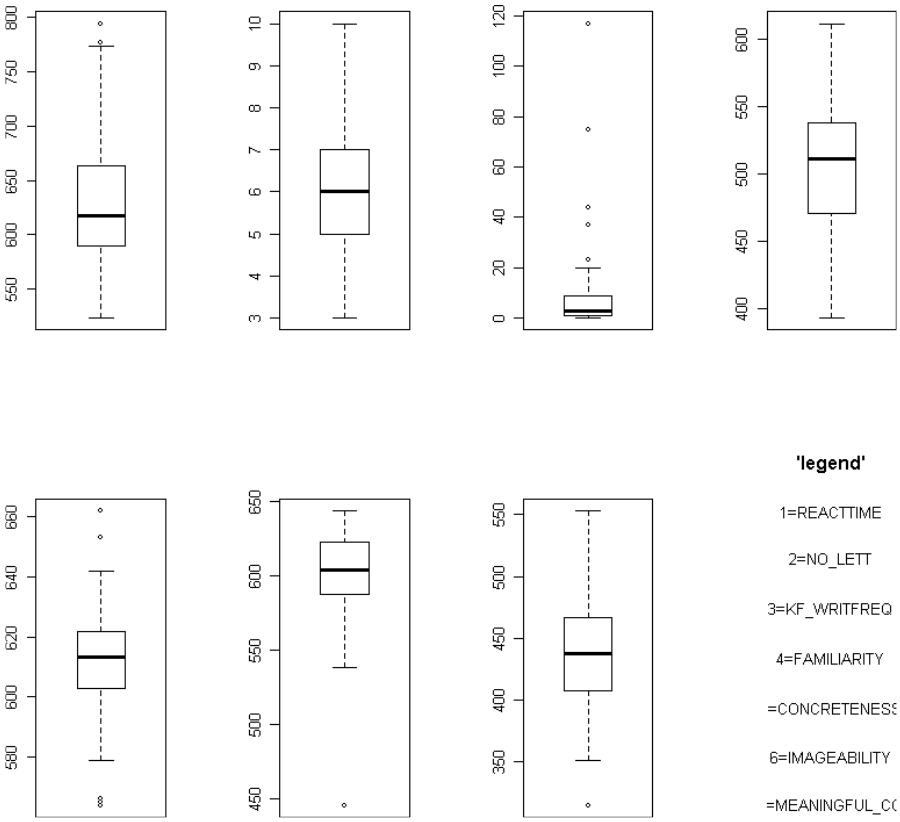


*Figure 60.* Boxplots for all columns in the data frame

The invisible thing is that the data structure `boxplot.output` now contains a lot of statistics. This data structure is a list, a structure mentioned before very briefly in Section 4.1.1.2. I will not discuss this data structure here in great detail, suffice it here to say that many functions in R use lists to store their results (as does `boxplot`) because this data structure is very flexible in that it can contain many different data structures of all kinds.

In the present case, the list contains seven lists, one for each boxplot (enter `str(boxplot.output)¶` to see for yourself). Each of the lists contains two matrices and four vectors, which contain information on the basis

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

of which R plotted the boxplots. What we are interested in here is the fourth element of each list, which is called `out` and contains what R considered outliers. How do you get the fourth element of each of the seven lists in the list `boxplot.output`? You can either do this manually for each list with subsetting. The only thing you need to know is that parts of lists are usually not extracted with single square brackets but double square brackets. Thus, this is how you get the outliers from the first group:

```
> boxplot.output[[1]][[4]]¶
·gherkin····stork
776.6582·794.4522
```

The more elegant alternative is this, but don't dwell long on how this works for now – read up on `sapply` when you're done with this chapter; I only show the function call.

```
> sapply(boxplot.output,·"[",·4)¶
```

The data points you get here are each variable's outliers that are plotted separately in the boxplots and that, sometimes at least, stick out in the scatterplots. We will have to be careful to see whether or not these give rise to problems in the linear modeling process later.

Before we begin with the regression, a few things need to be done. First, you will need to tell R how the parameters of the linear regression are to be computed (more on that later). Second, you will want to disregard the incomplete cases (because R would otherwise do that later anyway) so you downsize the data frame to one that contains only complete observations (with the function `complete.cases`). Third, for reasons that will become more obvious below, all predictor variables are centered (with `scale` from Section 3.1.4). Then you can attach the new data frame and we can begin:

```
> options(contrasts=c("contr.treatment",·"contr.poly"))¶
> ReactTime<-ReactTime[complete.cases(ReactTime),]¶
> ReactTime.2<-ReactTime¶
> ReactTime.2[,-1]<-apply(ReactTime.2[,-1],·2,·scale,·
      scale=F)¶
> attach(ReactTime.2)¶
```

As before we use the function `lm`, but this time we list several independent variables. Recall, we want to test each independent variable, but also each pairwise interaction. Thankfully, you don't have to enter all interactions manually because there is a shorthand notation for that: if you put all

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

variables for which you want interactions into parentheses and add a "^*n*" (where *n* is an integer), then R will generate and test all interactions up to the level *n*. Thus, you can write this to start your work on the regressions (I add a `data=…` argument to the `lm` function, which is strictly speaking not necessary since we used `attach`, but it makes some plotting etc. below easier. I omit the call and all significance codes in the results):[38]

```
> model.1<-lm(REACTTIME ~ (CONCRETENESS + FAMILIARITY + IMAGEABILITY + KF_WRITFREQ + MEANINGFUL_CO + NO_LETT)
      ^2, data=ReactTime.2)
> summary(model.1)
[…]
Residuals:
    Min      1Q  Median      3Q     Max
-56.378 -16.013  -1.352  13.313  49.753

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 604.081415  7.258239  83.227  < 2e-16
CONCRETENESS                  0.174377  0.384053   0.454  0.65356
FAMILIARITY                  -0.318653  0.168723  -1.889  0.07015
IMAGEABILITY                 -0.111855  0.360045  -0.311  0.75853
KF_WRITFREQ                  -6.627089  7.046023  -0.941  0.35560
MEANINGFUL_CO                -0.177886  0.208846  -0.852  0.40213
NO_LETT                      12.371051  3.481681   3.553  0.00148
CONCRETENESS:FAMILIARITY     -0.008318  0.015268  -0.545  0.59052
CONCRETENESS:IMAGEABILITY     0.050307  0.021627   2.326  0.02807
CONCRETENESS:KF_WRITFREQ      0.662994  0.458249   1.447  0.15990
CONCRETENESS:MEANINGFUL_CO    0.020024  0.017726   1.130  0.26896
CONCRETENESS:NO_LETT          0.163612  0.237059   0.690  0.49620
FAMILIARITY:IMAGEABILITY     -0.007331  0.007790  -0.941  0.35528
FAMILIARITY:KF_WRITFREQ       0.493292  0.245883   2.006  0.05534
FAMILIARITY:MEANINGFUL_CO     0.004569  0.005688   0.803  0.42912
FAMILIARITY:NO_LETT           0.120892  0.118363   1.021  0.31649
IMAGEABILITY:KF_WRITFREQ     -0.114299  0.517466  -0.221  0.82691
IMAGEABILITY:MEANINGFUL_CO   -0.022628  0.009809  -2.307  0.02928
IMAGEABILITY:NO_LETT         -0.429974  0.227784  -1.888  0.07028
KF_WRITFREQ:MEANINGFUL_CO     0.046693  0.274243   0.170  0.86612
KF_WRITFREQ:NO_LETT          -4.959232  3.472913  -1.428  0.16520
MEANINGFUL_CO:NO_LETT         0.029051  0.185781   0.156  0.87695
---
Residual standard error: 33.55 on 26 degrees of freedom
Multiple R-squared: 0.6942,    Adjusted R-squared: 0.4472
F-statistic: 2.811 on 21 and 26 DF,  p-value: 0.006743
```

A lot of information … Let us begin at the bottom. There is a very significant correlation of intermediate strength: multiple $R^2$ = 0.4472;

38. So far, we always tested the assumptions of a test before we actually did it. However, since testing the appropriateness of a linear regression requires values that you only get from it, we compute the regression first and then evaluate its appropriateness.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

$F = 2.811$; $df = 21, 26$; $p = 0.0067$.[39] We ignore the residual standard error and turn to the coefficients.[40] In the middle of the output under "Coefficients", you can see the results for every variable and every interaction. Some variables and interactions reach the standard level of significance, but most do not. Many publications now provide this table in their results section, but we will use a different strategy. Following Occam's razor, we will now in a stepwise fashion eliminate variables/interactions that are neither significant nor involved in significant interactions because, since they are not significant, their influence on REACTTIME is probably only random anyway and they must be omitted from the subsequent models.

As a first step, let us exclude the least significant interaction: MEANINGFULNESS:NUMBERLETTERS ($p \approx 0.88$). Thankfully, you don't have to now spell out the complete model – you can use `update`:

```
> model.2<-update(model.1,·~.·-·MEANINGFUL_CO:NO_LETT)¶
```

This tells R to create a new linear model, `model.2`, which is the same as `model.1` (that's what `model.1,·~.` means), but does *not* contain (hence the minus) the specified interaction. Let's look at the new model (I now only provide the coefficients, the $R^2$-values, and the overall significance test.)

```
> summary(model.2)¶
[…]
(Intercept)···············604.427392···6.786756··89.060··<·2e-16
CONCRETENESS···············0.162072···0.369051···0.439··0.66404
FAMILIARITY···············-0.320052···0.165413··-1.935··0.06355
IMAGEABILITY··············-0.125755···0.342538··-0.367··0.71639
KF_WRITFREQ···············-6.616024···6.917212··-0.956··0.34733
MEANINGFUL_CO·············-0.175571···0.204522··-0.858··0.39820
NO_LETT··················12.293915···3.383722···3.633··0.00116
CONCRETENESS:FAMILIARITY····-0.007639···0.014370··-0.532··0.59936
CONCRETENESS:IMAGEABILITY····0.049802···0.020995···2.372··0.02507
CONCRETENESS:KF_WRITFREQ·····0.689974···0.416785···1.655··0.10941
CONCRETENESS:MEANINGFUL_CO···0.018903···0.015918···1.188··0.24535
CONCRETENESS:NO_LETT·········0.186387···0.183630···1.015··0.31911
FAMILIARITY:IMAGEABILITY····-0.007115···0.007526··-0.945··0.35284
```

39. We use the second, adjusted $R^2$ value. The first one has the undesirable characteristic that it can only get larger as you include additional independent variables. The adjusted value, on the other hand, takes into consideration not only the amount of explained variance, but also the number of independent variables used to explain this amount of variance by subtracting a small amount from the $R^2$-value, which effectively penalizes the inclusion of many irrelevant variables.
40. The residual standard error is the square root of the quotient of the residual sums of squares divided by the residual degrees of freedom (in R: `sqrt(sum(residuals(model.1)^2)/26)¶`); I will not discuss this any further.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
FAMILIARITY:KF_WRITFREQ·····0.488236···0.239304···2.040··0.05122
FAMILIARITY:MEANINGFUL_CO····0.004644···0.005564···0.835··0.41126
FAMILIARITY:NO_LETT·········0.131016···0.097279···1.347··0.18924
IMAGEABILITY:KF_WRITFREQ····-0.080299···0.461007··-0.174··0.86302
IMAGEABILITY:MEANINGFUL_CO··-0.022963···0.009398··-2.444··0.02136
IMAGEABILITY:NO_LETT·····-0.411003···0.189274··-2.171··0.03885
KF_WRITFREQ:MEANINGFUL_CO····0.022696···0.223142···0.102··0.91974
KF_WRITFREQ:NO_LETT·····-4.936940···3.406722··-1.449··0.15880
[…]
Multiple·R-squared:·0.6939,·····Adjusted·R-squared:·0.4672
F-statistic:·3.061·on·20·and·27·DF,··p-value:·0.003688
```

What has happened now that a non-significant interaction has been removed? First, multiple $R^2$ is smaller, but only a tiny little bit – 0.0003 – which is not surprising since we dropped only an insignificant interaction from the model. Second and more interestingly, adjusted $R^2$ is larger and the *p*-value has decreased to nearly half the first value, again because we dropped an interaction without losing much predictive power. Put differently, we were rewarded for dropping useless variables/interactions, which changes the degrees of freedom. Third, note that the *p*-values changed: IMAGEABILITY:NUMBERLETTERS was marginally significant in `model.1` ($p = 0.07028$) but it is now significant ($p = 0.03885$). This is important because it shows that in such a multifactorial linear model, each predictor's effect is not evaluated in isolation but in the context/presence of the other predictors in the model: when one predictor is removed or added, everything else in the model can change.

Given that $R^2$ of `model.2` is nearly exactly as large as $R^2$ of `model.1`, the models don't seem to differ significantly from each other, but let us also test that. We can use the function `anova` for that, which for this application just takes the two models as arguments (where one model is a subset of the other): (I again omit the model definitions from the output.)

```
> anova(model.1,·model.2)¶
Analysis·of·Variance·Table
[…]
··Res.Df·····RSS·Df·Sum·of·Sq·····F·Pr(>F)
1·····26·29264.7
2·····27·29292.2·-1·····-27.5·0.0245··0.877
```

The *p*-value of 0.877 shows we were clearly justified in omitting the three interactions: there is no significant difference between the models so Occam's razor forces us to adopt the simpler one, `model.2`. And you may even recognize that *p*-value: it's the *p*-value of the interaction MEANINGFULNESS:NUMBERLETTERS, which we deleted from the model.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

Let's move on, there are still many insignificant predictors to delete and you delete predictors in a stepwise fashion and from highest-order interactions to lower-order interactions to main effects. Thus, we chose the next most insignificant one, KF-WRITTENFREQ: MEANINGFULNESS. Since there will be quite a few steps before we arrive at the minimal adequate model, I now often provide only very little output; you will see the complete output when you run the code.

```
> model.3<-update(model.2,·~.·-·KF_WRITFREQ:MEANINGFUL_CO)¶
> summary(model.3);·anova(model.2,·model.3)¶
[…]
Multiple·R-squared:·0.6938,·····Adjusted·R-squared:·0.486
F-statistic:·3.339·on·19·and·28·DF,··p-value:·0.001926

Analysis·of·Variance·Table
[…]
··Res.Df·····RSS·Df·Sum·of·Sq·····F·Pr(>F)
1·····27·29292.2
2·····28·29303.4·-1·····-11.2·0.0103·0.9197
```

Same story: multiple $R^2$ decreased a little, adjusted $R^2$ increased a little, $p$ decreased. The interaction of CONCRETENESS and KF-WRITTENFREQ changed to 'marginally significant', NUMBERLETTERS changed to highly significant, but not much else has happened. Again, `anova` shows that the model simplification was justified: $p \approx 0.92$.

And we go on like this, always eliminating the interaction with the largest $p$-values with `update` and checking that with `anova` … Follow along on the basis of the code file and see how adjusted $R^2$ changes, how variables' and interactions' $p$-values change, and how the ratio of significant predictors in our model increases:

```
> model.4<-update(model.3,·~.·-·IMAGEABILITY:KF_WRITFREQ)¶
> summary(model.4);·anova(model.3,·model.4)¶
> model.5<-update(model.4,·~.·-·CONCRETENESS:FAMILIARITY)¶
> summary(model.5);·anova(model.4,·model.5)¶
> model.6<-update(model.5,·~.·-·FAMILIARITY:MEANINGFUL_CO)¶
> summary(model.6);·anova(model.5,·model.6)¶
> model.7<-update(model.6,·~.·-·FAMILIARITY:IMAGEABILITY)¶
> summary(model.7);·anova(model.6,·model.7)¶
> model.8<-update(model.7,·~.·-·CONCRETENESS:NO_LETT)¶
> summary(model.8);·anova(model.7,·model.8)¶
> model.9<-update(model.8,·~.·-·FAMILIARITY:NO_LETT)¶
> summary(model.9);·anova(model.8,·model.9)¶
> model.10<-update(model.9,·~.·-·CONCRETENESS:KF_WRITFREQ)¶
> summary(model.10);·anova(model.9,·model.10)¶
> model.11<-update(model.10,·~.·-·KF_WRITFREQ:NO_LETT)¶
> summary(model.11);·anova(model.10,·model.11)¶
> model.12<-update(model.11,·~.·-·CONCRETENESS:IMAGEABILITY)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> summary(model.12); anova(model.11, model.12)¶
> model.13<-update(model.12, .~. - IMAGEABILITY:NO_LETT)¶
```

Now an interesting situation arises: This is the first time all interactions that are still in the model are at least marginally significant:

```
> summary(model.13); anova(model.12, model.13)¶
[…]
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              607.675439   5.600854 108.497  < 2e-16
CONCRETENESS              -0.080520   0.267324  -0.301 0.764898
FAMILIARITY               -0.265266   0.152372  -1.741 0.089792
IMAGEABILITY              -0.232681   0.273245  -0.852 0.399800
KF_WRITFREQ               -5.349738   5.154223  -1.038 0.305861
MEANINGFUL_CO             -0.008723   0.180080  -0.048 0.961617
NO_LETT                   11.575720   2.869773   4.034 0.000256
CONCRETENESS:MEANINGFUL_CO  0.020318   0.007844   2.590 0.013529
FAMILIARITY:KF_WRITFREQ     0.312600   0.130241   2.400 0.021393
IMAGEABILITY:MEANINGFUL_CO -0.008990   0.003435  -2.617 0.012657
---
[…]
Multiple R-squared: 0.5596,     Adjusted R-squared: 0.4553
F-statistic: 5.364 on 9 and 38 DF,  p-value: 9.793e-05

Analysis of Variance Table
[…]
  Res.Df   RSS Df Sum of Sq     F Pr(>F)
1     37 40678
2     38 42151 -1     -1473 1.3396 0.2545
```

You may now think that we go on removing non-significant main effects (such as MEANINGFULNESS) from the regression model until all predictors are significant. However, while one in general weeds out non-significant predictors in the way we did, recall from above that one does *not* weed out non-significant predictors that still participate in higher interactions. In this case, MEANINGFULNESS is the least significant predictor so you may want to remove it – but you don't because MEANINGFULNESS still participates in a significant interaction. In fact, if you look at all non-significant variables, you will find that this is true of all of them, which means that we cannot simplify the model any further and that `model.13` is our final, minimally adequate model, which indicates that there is an overall correlation that is highly significant.

Now, how well does this regression equation predict the observed reaction times? You can assess the model on the basis of adjusted $R^2$, but you can also check the exact predictions. Of course, you can generate a regression equation as in the sections on correlations above – an interaction of two ratio-scaled variables is by default modeled as the product of the two

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

variables' values for each data point – but this becomes very tedious so you use `predict` again:

```
> head(predict(model.13))·#·or·head(fitted(model.13))¶
·····ant·····apple·asparagus····banana·····bat····beaver
·581.4894··577.0885··645.0601··582.2992··598.7205··646.0908
```

And, as mentioned above in Section 3.2, you can also use `predict` to make predictions for combinations of values that were not observed. If you wanted to predict the value for the first word (this was of course observed, this is just so that you can check you get the right output), you specify the desired variable values in a list called `newdata`:

```
> predict(model.13,·newdata=list(NO_LETT=-2.625,·KF_WRITFREQ=
      0.1089857,·FAMILIARITY=-5.208333,·CONCRETENESS=
      -7.645833,·IMAGEABILITY=10.85417,·MEANINGFUL_CO=
      -20.97917))¶
.......1
581.4894
```

Let us briefly have a look at which words' reaction times are predicted well and which are not. The first of the following two lines sets up an empty coordinate system. (I set the limits of the *y*-axis manually so that all residuals can be shown and that the *y*-axis extends in both directions symmetrically around 0.) The second line plots the words at the *x*-axis values 1 to 8 (which also means, the position of a word on the *x*-axis does not mean anything: words are just spread out to avoid overplotting). Other and maybe nicer ways to plot this are shown in the code file.

```
> plot(1:8,·xlim=c(0,·9),·ylim=c(-100,·100),·xlab="",·
      ylab="Residuals·in·ms",·type="n");·grid()¶
> text(rep(1:8,·6),·residuals(model.17),·labels=
      row.names(ReactTime.2),·cex=0.9)¶
```

Obviously, the reaction times for the words *squirrel*, *potato*, *asparagus*, and *tortoise* are underestimated while the reaction times for the words *sheep*, *spider*, *apple*, and *orange*, for instance, are strongly overestimated, which could be explored further depending on the study's objectives. One thing worth mentioning, though, is that the words whose reaction times are not predicted well are not all exactly ones that looked like outliers in the variable-specific boxplots earlier. One of the words that appeared to be an outlier that would potentially bias the results (*horse*) is predicted rather well. Thus, the practice of simply excluding some high or low values (e.g.,

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

because they are two or three standard deviations away from the mean) can exclude data from consideration that can be accounted for very well. We will look at more appropriate ways to identify outliers below.

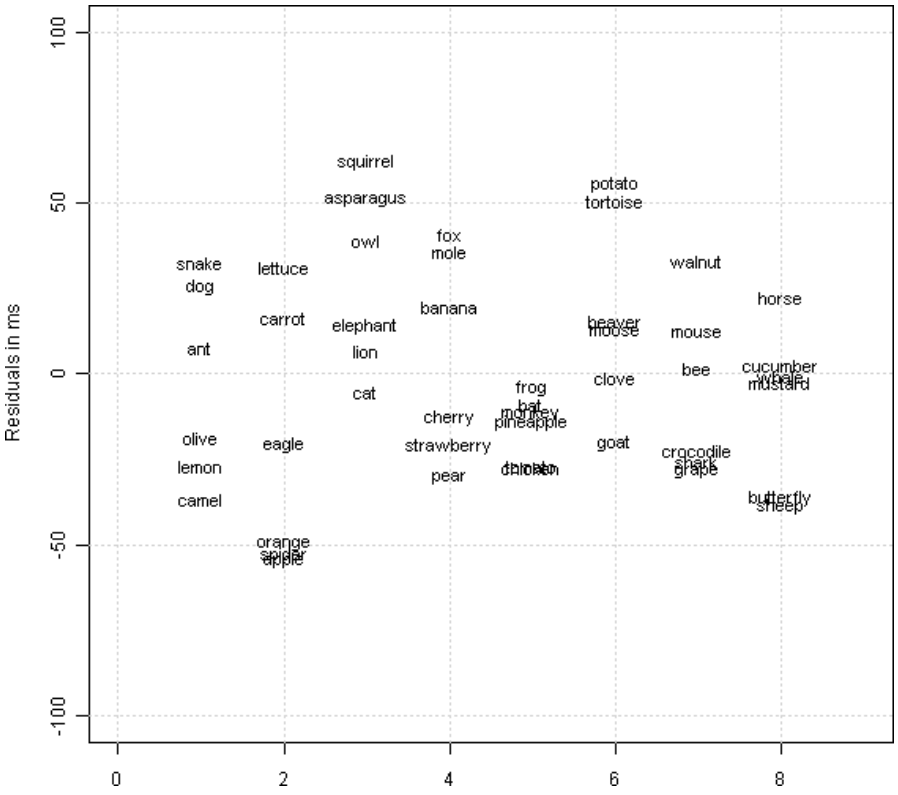


*Figure 61.* Residuals for all words

Some further explanations/comments: first, multiple $R^2$ corresponds to the squared correlation of the observed and predicted values, as you can easily verify by entering `cor(REACTTIME,·predict(model.13))^2¶`.

Second, the coefficients in the linear regression are of course only estimates, which is why it is always useful to also look at their confidence intervals to get a better idea of their precision. You don't have to compute them manually as in Section 3.1.5, but can use the function `confint`, which only requires the linear model as its only argument (for the default 95% confidence interval, that is – for a 99%-confidence interval, you add `level=0.99`).

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> confint(model.13)¶
·····································2.5·%·····97.5·%
(Intercept)·····················596.337102767·619.013775682
CONCRETENESS·····················-0.621690201···0.460649732
FAMILIARITY·····················-0.573727338···0.043195748
IMAGEABILITY·····················-0.785836257···0.320474310
KF_WRITFREQ·····················-15.783916233···5.084440221
MEANINGFUL_CO·····················-0.373276344···0.355829352
NO_LETT·····························5.766167271··17.385272293
CONCRETENESS:MEANINGFUL_CO···0.004437865···0.036197360
FAMILIARITY:KF_WRITFREQ·····0.048939463···0.576259695
IMAGEABILITY:MEANINGFUL_CO··-0.015943219··-0.002036064
```

But now what do the coefficients (which were computed using centered predictors, remember?) and their confidence intervals mean? In this case here, the coefficients of the main effects correspond to the predictive difference a variable makes with the other predictors in the model at their average values. Why is that so? This is so because we used centered predictors in our regression, which makes sure that the mean of the previously uncentered raw predictors is now 0. In fact, this is one of two reasons why we centered them: if you do not center variables this way, the coefficients are harder to interpret and are sometimes not particular meaningful (cf. below for an example). The second reason is that centering predictors protects a bit against what is called *collinearity* of predictors, the undesirable phenomenon that sp,e predictors may be correlated with each other, which can affect the coefficients and the power of the analysis. While this is a problem to large to be discussed here, the present data set in its raw form suffers from high collinearity whereas the centered form does not.

Thus, when all other variables are at their average, then a one-letter increase of a word increases the predicted reaction time by 11.576 ms. When all other variables are at their average, then an increase of one unit of FAMILIARITY decreases the predicted reaction time by 0.265 ms. From that, can you guess what the coefficient for the intercept actually is?

**THINK**
**BREAK**

Here's the answer in R:

```
> predict(model.13,·newdata=list(NO_LETT=0,·KF_WRITFREQ=0,·
      FAMILIARITY=0,·CONCRETENESS=0,·IMAGEABILITY=0,·
      MEANINGFUL_CO=0))¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
.......1
607.6754
```

The coefficient for the intercept is the predicted reaction time when each predictors is at its average. (And if we had not centered the predictors, the coefficient for the intercept would be the predicted reaction time when all variables are zero, which is completely meaningless here.)

For the coefficients of interactions, the logic is basically the same. Let's look at CONCRETENESS:MEANINGFULNESS, which had a positive coefficient, 0.020318. When both increase by 100, then, all other things being equal, they change the predicted reaction time by the sum of

- 100 · -0.080520 = -8.0520 (the main effect of CONCRETENESS);
- 100 · -0.008723 = -0.8723 (the main effect of MEANINGFULNESS);
- 100 · 100 · 0.020318 = 203.18ms (their interaction).

Disregarding differences due to rounding, the sum of these values (194.2557) is what R outputs, too:

```
> means.everywhere<-predict(model.13,·newdata=
      list(NO_LETT=0,·KF_WRITFREQ=0,·FAMILIARITY=0,·
      CONCRETENESS=0,·IMAGEABILITY=0,·MEANINGFUL_CO=0));·
      means.everywhere¶

607.6754
> both.positive<-predict(model.13,·newdata=list(NO_LETT=0,·
      KF_WRITFREQ=0,·FAMILIARITY=0,·CONCRETENESS=100,·
      IMAGEABILITY=0,·MEANINGFUL_CO=100))¶
> both.positive-means.everywhere¶
.......1
194.2518
```

On the other hand, when both decrease by 100, then, all other things being equal, they change the prediction by the sum of

- -100 · -0.080520 = 8.0520 (the main effect of CONCRETENESS);
- -100 · -0.008723 = 0.8723 (the main effect of MEANINGFULNESS);
- -100 · -100 · 0.020318 = 203.18ms (their interaction).

```
> both.negative<-predict(model.13,·newdata=
      list(NO_LETT=0,·KF_WRITFREQ=0,·FAMILIARITY=0,·
      CONCRETENESS=-100,·IMAGEABILITY=0,·
      MEANINGFUL_CO=-100))
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> both.negative-means.everywhere
.......1
212.1005
```

Third, note that the sizes of the coefficients in the regression equation do *not* reflect the strengths of the effects. These values have more to do with the different scales on which the variables are measured than with their importance. You must also not try to infer the effect sizes from the *p*-values. Rather, what you do is you compute the linear model again, but this time not on the centered values, but on the standardized values of both the dependent variable and all predictors, i.e., the columnwise *z*-scores of the involved variables and interactions (i.e., you will need `scale` again but this time with the default setting `scale=T`). For that, you need to recall that interactions in this linear regression model are products of the interacting variables. However, you cannot simply write the product of two variables into a linear model formula because the asterisk you would use for the product already means something else, namely 'all variables in isolation and all their interactions'. You have to tell R something like 'this time I want the asterisk to mean mere multiplication', and the way to do this is with by putting the multiplication in brackets and prefixing it with `I`. Thus:

```
> model.13.effsiz<-lm(scale(REACTTIME)·~·
       scale(FAMILIARITY)·+·scale(NO_LETT)·+·
       I(scale(MEANINGFUL_CO)*scale(CONCRETENESS))·+·
       I(scale(FAMILIARITY)*scale(KF_WRITFREQ))·+·
       I(scale(MEANINGFUL_CO)*scale(IMAGEABILITY)))¶
> round(coef(model.13.effsiz),·2)¶
·····································(Intercept)
·····································-0.13
·····························scale(CONCRETENESS)
·····································-0.04
·····························scale(FAMILIARITY)
·····································-0.28
·····························scale(IMAGEABILITY)
·····································-0.17
·····························scale(KF_WRITFREQ)
·····································-0.14
·····························scale(MEANINGFUL_CO)
·····································-0.01
·····························scale(NO_LETT)
·····································0.48
I(scale(CONCRETENESS)·*·scale(MEANINGFUL_CO))
·····································0.41
···I(scale(FAMILIARITY)·*·scale(KF_WRITFREQ))
·····································0.40
I(scale(IMAGEABILITY)·*·scale(MEANINGFUL_CO))
·····································-0.28
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

These coefficients, which are sometimes called *standardized regression coefficients* or *beta weights*, are one, though not completely unproblematic, way of assessing the degree of importance of each predictor left in the model. These standardized coefficients range from -1 to 1 and can be interpreted like correlation coefficients: positive and negative values reflect positive and negative correlations respectively (cf. the code file for a graphical representation of the effect sizes). The results from the above pairwise scatterplot are confirmed: CONCRETENESS and MEANINGFULNESS exhibited virtually no trends and have standardized coefficients very close to zero, and NUMBERLETTERS exhibited a positive trend and has a positive standardized coefficient. Note, however, that you cannot simply interpret all beta weights in this straightforward way because the interactions show that each of these variables participates in interactions that make it necessary to qualify their effect.

Consider, for example, again the interaction CONCRETENESS:MEANINGFULNESS. As you can see, as main effects, both variables contribute very little to the predictive power of the model: they are in fact the two weakest predictors as their beta weights are closest to zero. On the other hand, their interaction is the second strongest predictor in the whole model. One quick and dirty way to explore this interaction consists of dichotomizing the two variables (and since we centered all variables, we can dichotomize with 0 as the boundary) and then determine the means of the four possible combinations of the (then) two variable levels of the two then dichotomized variables. Again, since interactions are often easier to understand from one perspective, we immediately generate both tables:

```
> CONC.MEAN.1<-tapply(predict(model.13),·
       list(CONCRETENESS>0,·MEANINGFUL_CO>0),·mean);·
       CONC.MEAN.1¶
·········FALSE·····TRUE
FALSE·633.1282·601.5534
TRUE··610.3415·603.3445

> CONC.MEAN.2<-tapply(predict(model.13),·
       list(MEANINGFUL_CO>0,·CONCRETENESS>0),·mean);·
       CONC.MEAN.2¶
·········FALSE·····TRUE
FALSE·633.1282·610.3415
TRUE··601.5534·603.3445
```

What does this show?

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

**THINK**
**BREAK**

It shows that when MEANINGFULNESS is larger than average, CONCRETENESS being smaller or larger than average does not make much of a difference: the predicted means differ only by not even two ms. However, when MEANINGFULNESS is smaller than average, then values of CONCRETENESS that are also smaller than average increase the predicted reaction time much more; if you use the code in the code file to create barplots you will see the effect immediately.

Let us now finally look at the model assumptions. The standard check of the model assumptions involves four plots so you instruct R to create a two-by-two grid and then plot the model object (if you add the argument `which=1:6`, you get two more plots and must adjust the grid accordingly):

```
> par(mfrow=c(2,·2))¶
> plot(model.13)¶
> par(mfrow=c(1,·1))·#·restore·the·standard·plotting·setting¶
```

Consider Figure 62. What do these graphs mean? The two left graphs test the assumptions that the variances of the residuals are constant. Both show the ratio of the fitted/predicted values on the *x*-axis to the residuals on the *y*-axis (as the raw or the root of the standardized residuals). Ideally, both graphs would show a scattercloud without much structure, especially no structure such that the dispersion of the values increases or decreases from left to right. Here, both graphs look good.[41] Several words – *potato*, *squirrel*, and *apple*/*tortoise* – are marked as potential outliers. Also, the plot on the top left shows that the residuals are distributed well around the desired mean of 0.

The assumption that the residuals are distributed normally also seems met: The points in the top right graph should be rather close to the dashed line, which they are; again, three words are marked as potential outliers. But you can of course also do a Shapiro-Wilk test on the residuals, which yields the result hoped for.

41. You can also use `ncv.test` from the `library(car)`: `library(car);·ncv.test(model.13)¶`, which returns the desired non-significant result.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

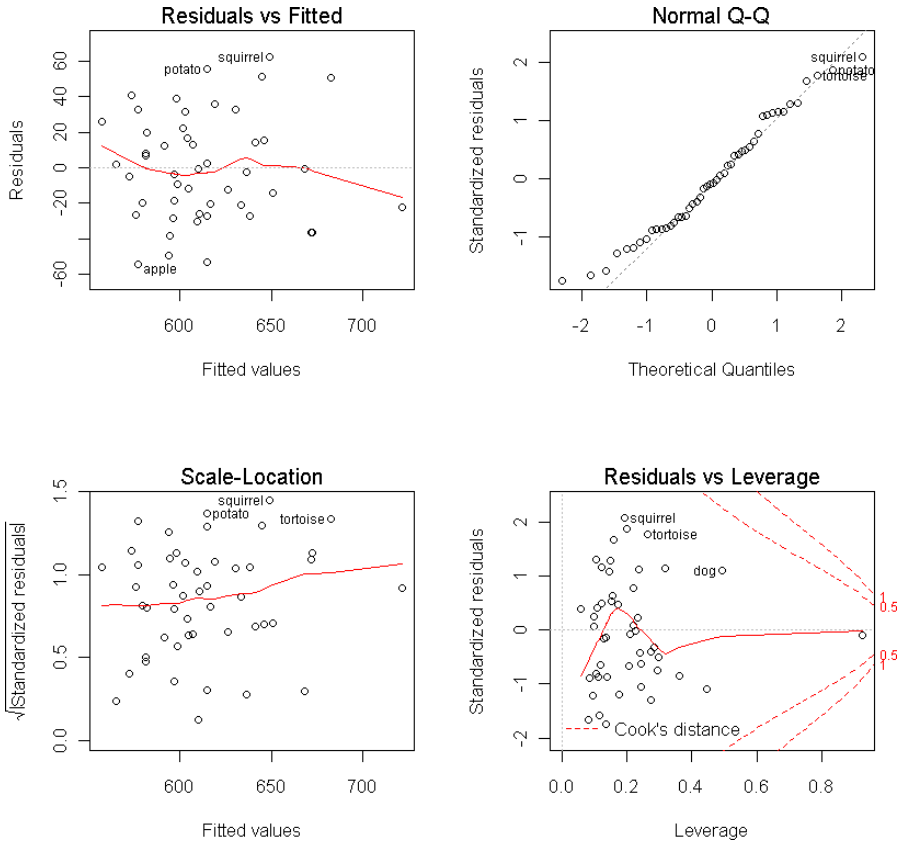


*Figure 62.* Model-diagnostic plots

```
> shapiro.test(residuals(model.13))¶
        Shapiro-Wilk normality test
data:  residuals(model.13)
W = 0.9769, p-value = 0.458
```

Finally, the bottom right plot plots the standardized residuals against the so-called leverage. Leverage is a measure of how much a point can influence a model. (Note: this is not the same as outliers, which are points that are not explained well by a model such as, here, *squirrel*.) You can compute these leverages most easily with the function `hatvalues`, which only requires the fitted linear model as an argument. (In the code file, I also show you how to generate a simple plot with leverages.)

```
> hatvalues(model.13)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

As you could see, there is only one word with a very large leverage, *clove*, which is why we do not need to worry about this too much (recall from above that *clove* was the word that turned up several times as an outlier in the boxplot). One thing you might want to try, also, is fit the model again without the word *squirrel* since each plot in the model diagnostics has it marked as a point that has been fitted rather badly. Let's see what happens:

```
> model.13.wout.squirrel<-lm(formula(model.13),·
      data=ReactTime.2[-43,])¶
> summary(model.13.wout.squirrel)¶
[…]
Coefficients:
·····························Estimate·Std.·Error·t·value·Pr(>|t|)
(Intercept)·····················607.107405···5.346363·113.555··<·2e-16
CONCRETENESS····················-0.107829···0.255182··-0.423·0.675065
FAMILIARITY·····················-0.257604···0.145320··-1.773·0.084517
IMAGEABILITY····················-0.402052···0.271746··-1.480·0.147467
KF_WRITFREQ·····················-4.541263···4.928068··-0.922·0.362754
MEANINGFUL_CO···················0.090203···0.177531···0.508·0.614401
NO_LETT·························10.765275···2.761037···3.899·0.000392
CONCRETENESS:MEANINGFUL_CO···0.018404···0.007530···2.444·0.019406
FAMILIARITY:KF_WRITFREQ·····0.273133···0.125477···2.177·0.035953
IMAGEABILITY:MEANINGFUL_CO··-0.008513···0.003282··-2.594·0.013523
[…]
Multiple·R-squared:·0.5655,·····Adjusted·R-squared:·0.4599
F-statistic:·5.352·on·9·and·37·DF,··p-value:·0.0001094
```

The coefficients do not that change much and both multiple $R^2$ and, more importantly, adjusted $R^2$ go up a bit, but ideally you would also look at the coefficients you get for the effect size model. If you do that (cf. the code file), you find that the largest change arises for IMAGEABILITY. The word *squirrel* was clearly responsible for some residual variance. Note that you can of course not eliminate all values you don't like – you must analyze the data carefully to justify the elimination of data points.

Let us now sum up the results:[42] "A linear regression was used to study the effects of NUMBERLETTERS, KF-WRITTENFREQ, FAMILIARITY,

42. A short comment is still necessary: the example above may not be ideal because the range of the dependent variable is limited: reaction times can, for example, not be negative but the regression equation may well predict negative values. It is important to bear in mind that the regression equation's predictive power is best only for the range of observed values. Other kinds of regression are sometimes recommended to deal with such cases because their link functions restrict the range of predicted values. For example, Poisson regressions only predict positive values (cf. Faraway 2006: Ch. 3 and Crawley 2007: Ch. 14, 16 for discussion of Poisson regression and regressions involving percentages).

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

CONCRETENESS, IMAGEABILITY, and MEANINGFULNESS and their pairwise interactions on the reaction times of 48 words in lexical decision tasks (29 incomplete cases were discarded). A first model with all independent variables and their pairwise interactions was fit. In a stepwise procedure, then the variable or interaction with the highest (insignificant) *p*-value was deleted until only variables and interactions participating in significant predictors remained. This model contained all six independent variable, as well as the significant interactions CONCRETENESS:MEANINGFULNESS, FAMILIARITY:KF-WRITTENFREQ, and IMAGEABILITY:MEANINGFULNESS. Model diagnostics showed that the assumptions of linear regressions were not violated, but that one word, *squirrel*, was marked as an outlier in several tests. The above model resulted in a highly significant overall correlation (adjusted $R^2 = 0.4553$; $F = 5.364$; $df = 9, 38$; $p < 0.001$) with the following parameters: [show the table of coefficients from `model.13` and graphs]. In this model, the variable NUMBERLETTERS has the strongest effect (and increases reaction times) whereas MEANINGFULNESS and CONCRETENESS as main effects have the weakest effects [show the table of coefficients from `model.13.effsiz` and maybe graphs with beta-weights]. The five words that are predicted worst in this model are *squirrel*, *potato*, *apple*, *spider*, and *asparagus*. and the word that has the strongest effect on the model parameters is *clove*. [… plus your interpretation]"

**Recommendation(s) for further study**

- the function `step`, to have R perform model selection automatically based on criteria other than the coefficients' *p*-values rather than do it with yourself with `anova`; cf. also the functions `leaps` and `regsubsets` (from the `library(leaps)`), the function `fastbw` (from the `library(Design)`), and the function `stepAIC` (from the `library(MASS)`). Warning: automatic model selection processes can be dangerous: they cannot include the analyst's expert knowledge and different algorithms can result in very (!) different results; just apply `step` and `fastbw` to this data set …
- the function `collin.fnc` from the `library(languageR)` to test for collinearity
- the function `gvlma` from the `library(gvlma)` to test model assumptions
- the function `calc.relimp` (from the `library(relaimpo)`), to compute measures of importance for variables in a linear model
- the functions `influence.measures`, `rstudent`, `lm.influence`, and `influence`, to identify outliers
- the functions `qplot` (from the `library(ggplot)`) and `coefplot` (from

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

the `library(arm)`) for nice plots with confidence intervals
- R also allows you to compute so-called robust regressions (`library(robust)`) and (`library(MASS)`) as well as nonlinear regressions (`library(nls)`), which we cannot deal with here; cf. esp. Crawley (2002, 2005) and Faraway (2005, 2006) for these (more complex) topics as well as Good and Hardin (2006: Ch. 11) for the `library(quantreg)` and further alternatives
- It is worth exploring the issue of collinearity, which refers to the threat that highly intercorrelated predictor variables pose for regression coefficients and their *p*-values; cf. Faraway (2005: Sections 5.3 and 9.3)
- Harrell (2001), Crawley (2002: Ch. 13, 14, 17), Backhaus et al. (2003: Ch. 1), Maindonald and Braun (2003: Ch. 3-6), Bortz (2005: Ch. 13), Crawley (2007: Ch. 10, 14, 16), Gelman and Hill (2007: Ch. 3-4), Baayen (2008: Ch. 6-7), Johnson (2008: Section 2.4, 3.2)
- Good and Hardin (2006: Ch. 10) on risks of linear regressions

### 3. ANOVA (analysis of variance)

In Section 4.3.2.1, I explained how you test whether arithmetic means from two independent samples are significantly different from each other. As an example, we looked at different F1 frequencies of men and women. The *t*-test for independent samples from that section, however, cannot be applied to cases where you need to compare more than two means (because your single independent variable has more than two levels or because you have more than one independent variable). For both such situations, one often uses a method called ANOVA, for analysis of variance. In Section 5.3.1, I will explain how to perform a monofactorial ANOVA, and Section 5.3.2 deals with a multifactorial ANOVA, i.e., an ANOVA with more than one independent variable; in the context of ANOVAs independent variables are often referred to as factors (which have nothing to do with factors in factor analysis).[43]

43 Let me also remind you that nominal/categorial variables are ideally coded (with strings) such that R's `read.table` can automatically recognize them (cf. Section 2.5.1). The independent variables can in fact also include ratio-scaled variables; sometimes, the method is then referred to as ANCOVA (analysis of covariance). The overall procedure is practically the same as with 'regular' ANOVAs.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

### 3.1. Monofactorial ANOVA

To get to know monofactorial ANOVAs, we will extend the example from Section 4.3.2.3. Above, we studied subtractive word-formation processes and tested whether the average similarity of words entering into blends is different from that of words entering into complex clippings. If you want to include a third group of source words, the *t*-test or the *U*-test are not applicable anymore. In such a case, our question involves

- a dependent ratio-scaled variable, namely the similarity of the source words measured in DICE coefficients, whose average you are looking at;
- an independent categorical variable, namely PROCESS: *BLEND* vs. PROCESS: *COMPLCLIP* vs. PROCESS: *COMPOUND*.

The overall procedure of an ANOVA is this:

| **Procedure** |
|---|
| Formulating the hypotheses |
| Computing the means; inspecting graphical representations |
| Testing the main assumption(s) of the test: |
| the variances of the variable values in the groups are homogeneous and normally distributed in the populations from which the samples were taken or, at least, in the samples themselves |
| the residuals are normally distributed (with a mean of 0) in the populations from which the samples were taken or, at least, in the samples themselves |
| Computing the multiple correlation $R^2$ and the differences of means |
| Computing the test statistic *F*, the degrees of freedom *df*, and the probability of error *p* |

As usual, you begin with the hypotheses, which will be simplified below:

$H_0$: The means of the Dice coefficients of the source words entering into the three kinds of word-formation processes do not differ from each other: $mean_{\text{Dice coefficients of the blends}} = mean_{\text{Dice coefficients of the complex clippings}} = mean_{\text{Dice coefficients of the compounds}}$.

$H_1$: There is at least one difference between the average Dice coefficients of the three word-formation processes: $H_0$, with at least one "≠" instead of a "=".

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

You first clear R's memory and then load the data from <C:/_sflwr/_inputfiles/05-3-1_dices.txt>.

```
> rm(list=ls(all=T))¶
> Dices<-read.table(choose.files(),·header=T,·sep="\t",·
      comment.char="",·quote="")¶
> attach(Dices);·str(Dices)¶
'data.frame':···120·obs.·of··3·variables:
·$·CASE···:·int··1·2·3·4·5·6·7·8·...
·$·PROCESS:·Factor·w/·3·levels·"Blend","ComplClip",..:·3·...
·$·DICE···:··num··0.0032·0.0716·0.037·0.0256·0.01·0.0415·...
```

Then you compute the means and represent the data in a boxplot. To explore the graphical possibilities a bit more, you could also add the three means and the grand mean into the plot; I only provide a shortened version of the code here, but you will find the complete code in the code file for this chapter.

```
> boxplot(DICE~PROCESS,·notch=T,·ylab="Dices");·grid()¶
> text(1:3,·c(0.05,·0.15,·0.15),·labels=paste("mean=\n",·
      round(tapply(DICE,·PROCESS,·mean),·4),·sep=""))¶
> rug(DICE,·side=4)¶
> abline(mean(DICE),·0,·lty=2)¶
> text(1,·mean(DICE),·"Grand·mean",·pos=3,·cex=0.8)¶
```

The graph already suggests a highly significant result:[44] The Dice coefficients for each word-formation process appear to be normally distributed (the boxes and the whiskers are rather symmetric around the medians and means); the medians differ strongly from each other and are outside of the ranges of each others' notches. Theoretically, you could again be tempted to end the statistical investigation here and interpret the results, but, again, of course you can't really do that … Thus, you proceed to test the assumptions of the ANOVA.

The first assumption is that the variances of the variables in the groups in the population, or the samples, are homogeneous. This assumption can be tested with an extension of the *F*-tests from Section 4.2.2, the Bartlett-test. The hypotheses correspond to those of the *F*-test:

$H_0$: The variances of the Dice coefficients of the three word-formation processes are identical: $variance_{\text{Dice coefficients of blends}} = variance_{\text{Dice coefficients of complex}}$ clippings = $variance_{\text{Dice coefficients of compounds}}$.

44. Cf. the code file for this chapter for other graphs.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

$H_1$: The variances of the Dice coefficients of the three word-formation processes are not all identical: $H_0$, with at least one "≠" instead of a "=".

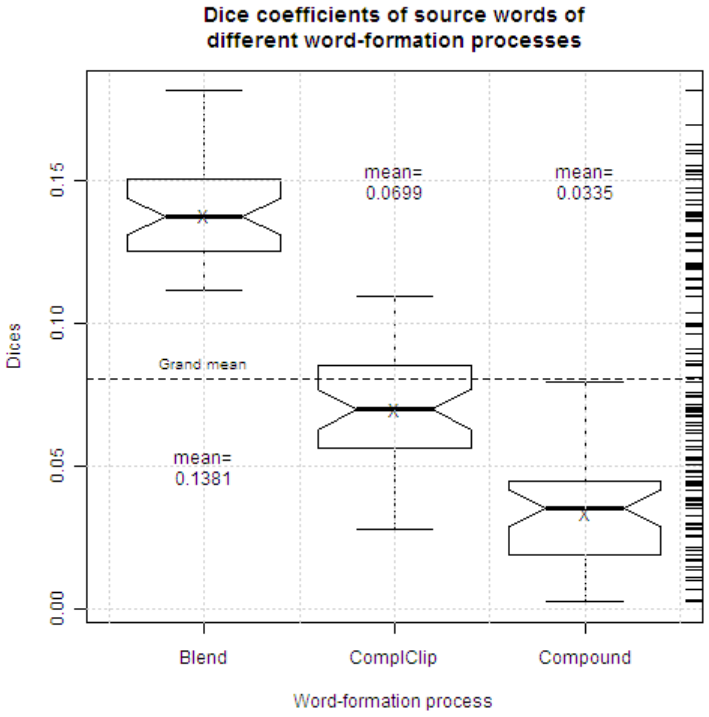


*Figure 63.* Boxplot for DICE~PROCESS

The standard deviations, i.e., the square roots, are very similar to each other:

```
> round(tapply(DICE, PROCESS, sd), 2)
   Blend ComplClip Compound
    0.02     0.02     0.02
```

In R, you can use the function `bartlett.test`. Just like most tests you have learned about above, you can use a formula and, unsurprisingly, the variances do not differ significantly from each other:

```
> bartlett.test(DICE~PROCESS)
        Bartlett test of homogeneity of variances
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
data:  DICE by PROCESS
Bartlett's K-squared = 1.6438, df = 2, p-value = 0.4396
```

The other assumption will be tested once the linear model has been created (just like for the regression).

A monofactorial ANOVA is based on an *F*-value which is the quotient of the variability in the data that can be associated with the levels of the independent variable divided by the variability in the data that remains unexplained. One implication of this is that the formulation of the hypotheses can be simplified as follows:

$H_0$: $F = 1$.
$H_1$: $F > 1$.

I will not discuss the manual computation in detail but will immediately turn to how you do this in R. For this, you will again use the functions `lm` and `anova`, which you already know. Again, you first define a contrast option that tells R how to compute the parameters of the linear model.[45]

```
> options(contrasts=c("contr.sum", "contr.poly"))¶
> model<-lm(DICE~PROCESS)¶
> anova(model)¶
Analysis of Variance Table
Response: DICE
           Df  Sum Sq  Mean Sq F value    Pr(>F)
PROCESS     2 0.225784 0.112892  332.06 < 2.2e-16 ***
Residuals 117 0.039777 0.000340
[...]
```

In the column "Df", you find the degrees of freedom for the independent variable PROCESS and for the residual variance that remains unexplained. In the column "Sum Sq", you find sums of squares, and the column "Mean Sq" contains the quotient $^{\text{Sum Sq}}/_{\text{Df}}$. The column "F value" contains the *F*-value, which is the quotient $^{0.112892}/_{0.00034}$. Finally, the column "Pr( > F)" lists the *p*-value for the *F*-value at 2 and 117 *df*. This *p*-value shows that the variable PROCESS accounts for a highly significant portion of the variance of the Dice coefficients. You can even compute how much

45. The definition of the contrasts that I use is hotly debated in the literature and in statistics newsgroups. (For experts: R's standard setting are treatment contrasts, but ANOVA results reported in the literature are often based on sum contrasts.) I will not engage in the discussion here which approach is better but, for reasons of comparability, will use the latter option and refer you to the discussion in Crawley (2002: Ch. 18, 2007: 368ff.) as well as lively repeated discussions on the R-help list.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

of the variance, namely the percentage of the sums of squares of PROCESS (0.225784) out of the complete sums of squares (0.225784+0.039777). The result of this is the effect size eta-squared $\eta^2$, an equivalent to the beta weight of the regression. With one independent variable, this is also $R^2$:

```
> 0.225784/(0.225784+0.039777)¶
[1] 0.8502152
```

This output does not reveal, however, which levels of PROCESS are responsible for significant amounts of variance. Just because PROCESS as a whole is significant, this does not mean that every single level of PROCESS has a significant effect (even though, here, Figure 63 suggests just that). A rather conservative way to approach this question involves the function `TukeyHSD`. The first argument of this function is an object created by the function `aov` (an alternative to `anova`), which in turn requires the relevant linear model as an argument; as a second argument you can order the order of the variable levels, which we will just do:

```
> TukeyHSD(aov(model), ordered=T)¶
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered
Fit: aov(formula = model)
$PROCESS
                      diff       lwr       upr p adj
ComplClip-Compound 0.0364675 0.02667995 0.04625505     0
Blend-Compound     0.1046600 0.09487245 0.11444755     0
Blend-ComplClip    0.0681925 0.05840495 0.07798005     0
```

The first two columns provide the observed differences of means; the columns "lwr" and "upr" provide the lower and upper limits of the 95% confidence intervals of the differences of means; the rightmost column lists the *p*-values (corrected for multiple testing; cf. Section 5.1.1 above) for the differences of means. You can immediately see that all means are highly significantly different from each other (as the boxplot already suggested).[46]

46. If not all variable levels are significantly different from each other, then it is often useful to lump together the levels that are not significantly different from each other and test whether a new model with these combined variable levels is significantly different from the model where the levels were still kept apart. If yes, you stick to and report the more complex model – more complex because it has more different variable levels – otherwise you adopt and report the simpler model. The logic is similar to the chi-square test exercises #3 and #13 in <04_all_exercises_answerkey.r>. The code file for this section discusses this briefly on the basis of the above data; cf. also Crawley (2007:364ff.).

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

For additional information, you can now also look at the summary of the linear model:

```
> summary(model)¶
[…]
Residuals:
·····Min·····1Q····Median·····3Q·····Max
-4.194e-02·-1.329e-02··1.625e-05··1.257e-02··4.623e-02

Coefficients:
···········Estimate·Std.·Error·t·value·Pr(>|t|)
(Intercept)··0.080512···0.001683··47.833···<2e-16·***
PROCESS1·····0.057617···0.002380··24.205···<2e-16·***
PROCESS1····-0.010575···0.002380··-4.443·2.03e-05·***
---
[…]
Residual·standard error:·0.01844·on·117·degrees·of·freedom
Multiple·R-squared:·0.8502,·····Adjusted·R-squared:·0.8477
F-statistic:·332.1·on·2·and·117·DF,··p-value:·<·2.2e-16
```

At the bottom, you see the overall significance test (with $F$, $df_{\text{PROCESS}}$, $df_{\text{residual}}$, and $p$). Above that, you find multiple $R^2$, which you already computed, as well as the adjusted version, which takes the amount of variables and variable levels involved into consideration. We again ignore the residual standard error and turn to the coefficients. When you use sum contrasts, as we do here (recall the `options` setting), then the intercept estimate is the overall mean of the dependent variable, and the rest of that row provides the test whether that grand mean is significantly different from 0. The following two lines list the differences of the means for the alphabetically first two levels of PROCESS. The value for blends – the alphabetically first level of PROCESS – is 0.057617 larger than the overall mean. The value for complex clippings – the alphabetically second level of PROCESS is 0.010575 smaller than the overall mean. The value for compounds – the alphabetically last and not listed level of PROCESS – differs from the overall mean by -0.057617+0.010575=-0.047042. These results correspond to those of Figure 63 and are also those returned by the function `model.tables`, which again requires `aov` and provides the results in a more accessible fashion:

```
> model.tables(aov(model))¶
Tables·of·effects
·PROCESS
PROCESS
····Blend·ComplClip·Compound
··0.05762··-0.01057·-0.04704
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

(Note in passing that, with `model.tables(aov(model),·"means")¶`, you get not the differences between means but the means themselves – try it out.) The additional columns of `summary(model)¶` provide the standard errors for the differences (in the first row for the difference of the blends' mean to 0, then for the differences between the blends' mean and the respective group means), and all these come with *t*-values and *p*-values. Above these coefficients, you find information about the residuals, which suggests that the residuals are distributed normally.

Let us now look at the model-diagnostic graphs. In this case, the four graphs do not indicate big problems. The plot on the top left shows that the residuals are just about perfectly distributed around the desired mean of 0. Both plots on the left do not reveal any particular structure, and the residuals in the upper right plot look as if they are normally distributed, which is confirmed by a Shapiro-Wilk test:

```
> par(mfrow=c(2,·2))¶
> plot(lm(DICE~PROCESS))¶
> par(mfrow=c(1,·1))·#·restore·the·standard·plotting·setting¶
> shapiro.test(residuals(model))¶
·····Shapiro-Wilk·normality·test
data:··lm(DICE·~·PROCESS)$residuals
W·=·0.9924,·p-value·=·0.7607
```

Three data points appear to be outliers:

```
> Dices[c(14,·71,·105),]¶
····CASE···PROCESS···DICE
14····14··Compound·0.0797
71····71·ComplClip·0.0280
105··105·····Blend·0.1816
```

As you can see, these three cases represent (i) the maximal Dice coefficient for the compounds, (ii) the minimal Dice coefficient for complex clippings, and (iii) the maximal Dice coefficient of all cases, which was observed for a blend; these values get also identified as extreme when you sort the residuals: `sort(residuals(model))¶`. One could now also check which word formations these cases correspond to.

Before we summarize the results, let me briefly also show one example of model-diagnostic plots pointing to violations of the model assumptions. Figure 65 below shows the upper two model plots that I once found when exploring the data of a student who had been advised by a stats consultant to apply an ANOVA to her data. In the left panel, you can clearly see how the range of residuals increases from left to right. In the right panel, you

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

can see how strongly the points deviate from the dashed line especially in the upper right part of the coordinate system. Such plots are a clear warning (and the function `gvlma` mentioned above showed that four out of five tested assumptions were violated!). One possible follow-up would be to see whether one can justifiably ignore the outliers indicated.

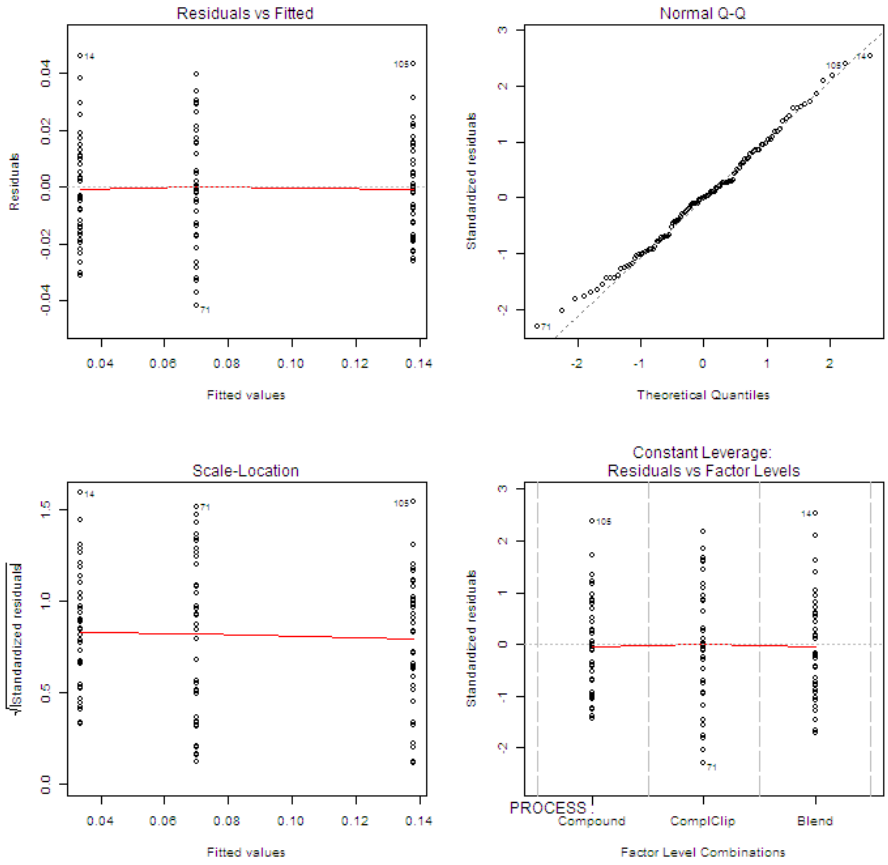


*Figure 64.* Model-diagnostic plots

After the evaluation, you can now summarize the analysis: "The similarities of the source words of the three word-formation processes as measures in Dice coefficients are very different from each other. The average Dice coefficient for blends is 0.1381 while that for complex clippings is only 0.0699 and that for compounds is only 0.0335 (all standard deviations = 0.02). [Then insert Figure 63.] According to a monofactorial ANOVA,

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

these differences are highly significant: $F_{2, 117} = 332.06$; $p < 0.001$ ***; the variable PROCESS explains more than 80% of the overall variance: multiple $R^2 = 0.85$; adjusted $R^2 = 0.848$. Pairwise *post-hoc* comparisons of the means (Tukey's HSD) show that all three means are significantly different from each other; all *p*s < 0.001." Again, it is worth emphasizing that Figure 63 already anticipated nearly all of these results; a graphical exploration is not only useful but often in fact indispensible.

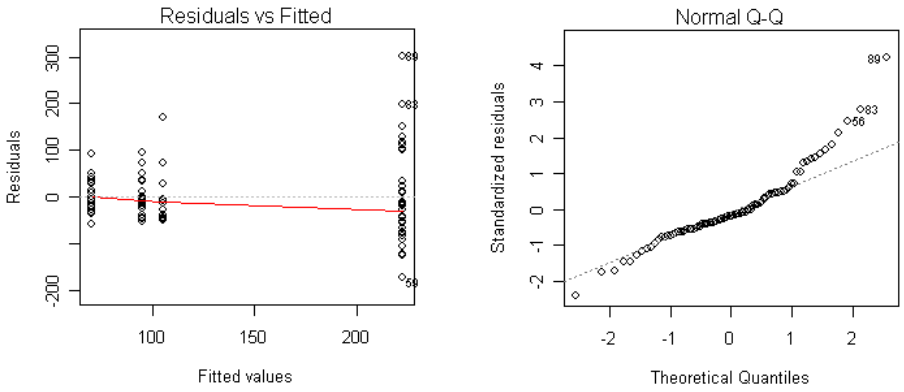


*Figure 65.* Problematic model-diagnostic plots

**Recommendation(s) for further studies (cf. also after Section 5.3.2)**

- the function `aov` already mentioned above is an alternative to `anova`
- the function `model.tables`, to get means (cf. above) and standard errors of a linear model
- the function `plot.design`, to plot monofactorial effects
- the function `oneway.test` as an alternative to monofactorial ANOVAs, which does not assume variance homogeneity
- the function `kruskal.test` as another non-parametric alternative to monofactorial ANOVAs
- the function `pairwise.t.test` as an alternative to `TukeyHSD` (unlike `TukeyHSD`, this function only works for monofactorial ANOVAs)
- Good and Hardin (2006: 71–73) on alternatives to ANOVAs
- (ANOVAs are often relatively robust when it comes to violations of their assumptions.)

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

## 3.2. Two-/multifactorial ANOVA

To explain this method, I will return to the example from Section 1.5. In that section, we developed an experimental design to test whether the numerical estimates for the meaning of some depends on the sizes of the quantified objects and/or on the sizes of the reference points used to locate the quantified objects. We assume for now the experiment resulted in a data set you now wish to analyze.[47] Since the overall procedure of ANOVAs has already been discussed in detail, we immediately turn to the hypotheses:

$H_0$: The average estimates in the four experimental conditions do not differ from each other: $mean_{\text{estimates for small objects}} = mean_{\text{estimates for large objects}} = mean_{\text{estimates for small reference points}} = mean_{\text{estimates for large reference points}} = mean_{\text{estimates for small objects and small reference points}} = mean_{\text{estimates for small objects and large reference points}} = mean_{\text{estimates for large objects and small reference points}} = mean_{\text{estimates for large objects and large reference points}}$.

$H_1$: There is at least one difference between the averages: there is at least one "≠" instead of an "=".

This time, you also immediately formulate the short version of the statistical hypotheses:

$H_0$: $F = 1$.
$H_1$: $F > 1$.

As the hypotheses indicate, you look both for main effects (i.e., each independent variable's effect in isolation) and an interaction (i.e., each independent variable's effects in the context of the other independent variable; cf. Section 1.3.2.3. You clear the memory, load the `library(car)`, and load the data from <C:/_sflwr/_inputfiles/05-3-2_objectestimates.txt>. You again use `summary` to check the structure of the data:

```
> rm(list=ls(all=T))¶
> library(car)¶
> ObjectEstimates<-read.table(choose.files(),·header=T,·
```

47. To make it easier to also check the results manually, I use a data set that does not contain the complete experimental design from Section 1.5. The overall logic of the analysis is of course the same as that of one based on a larger amount of data. Also, in order to keep things simple, I again do not address the issues of repeated measures and item-specific adjustments in the context of mixed effects / multi-level models but direct you to the references mentioned above.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
       sep="\t",·comment.char="",·quote="")¶
> attach(ObjectEstimates);·summary(ObjectEstimates)¶
·····CASE·····OBJECT···REFPOINT····ESTIMATE
·Min.···:·1.00···large:8···large:8···Min.···:·2.0
·1st·Qu.:·4.75···small:8···small:8···1st·Qu.:38.5
·Median·:·8.50·····Median·:44.0
·Mean···:·8.50·····Mean···:51.5
·3rd·Qu.:12.25·····3rd·Qu.:73.0
·Max.···:16.00·····Max.···:91.0
```

You can begin with the graphical exploration. In this case, where you have two independent binary variables, you can begin with the standard interaction plot, and as before you plot both graphs. The following two lines generate the kind of graph we discussed above in Section 3.2.2.2:

```
> interaction.plot(OBJECT,·REFPOINT,·ESTIMATE,·
       ylim=c(0,·90),·type="b")¶
> interaction.plot(REFPOINT,·OBJECT,·ESTIMATE,·
       ylim=c(0,·90),·type="b")¶
```

Also, you compute all means (cf. below) and all standard deviations (cf. the code file for this chapter):

```
> means.object<-tapply(ESTIMATE,·OBJECT,·mean)¶
> means.refpoint<-tapply(ESTIMATE,·REFPOINT,·mean)¶
> means.interact<-tapply(ESTIMATE,·list(OBJECT,·REFPOINT),·
       mean)¶
```

Instead of the standard interaction plots shown in Section 3.2.2.2, I show here two somewhat more sophisticated versions (in Figure 66). The code file for this chapter shows you how I generated this graph. As usual, it doesn't matter if you do not understand the code right away – once you try to understand the code and look up a few functions (such as `arrows`) you will learn a lot about how to do such graphs yourself.

Figure 66 shows both ways of looking at all means and their standard errors. As it is so often the case, the graph already anticipates virtually all of the results:

- The grand mean – the overall mean the dependent variable – is a bit larger than 50;
- In the middle of the left panel, you see the means for the levels of the variable REFPOINT when you disregard the levels of OBJECT while the left and right parts of the left panel show you the means of the interaction. Obviously, the means for the levels of REFPOINT will deviate sig-

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

nificantly from the grand mean since the middle error bars do not include the grand mean. On the whole, large reference points lead to larger estimates and small reference points lead to smaller estimates.

– In the middle of the right panel, you see the means for the levels of the variable OBJECT when you disregard the levels of REFPOINT while the left and right parts of the left panel show you the means of the interaction. Obviously, the means for the levels of OBJECT will not deviate significantly from the grand mean since the middle error bars do include the grand mean.
– Finally, Figure 66 strongly suggests that there is a significant interaction: the tendency that large reference points result in higher estimates seems to hold only for small objects.

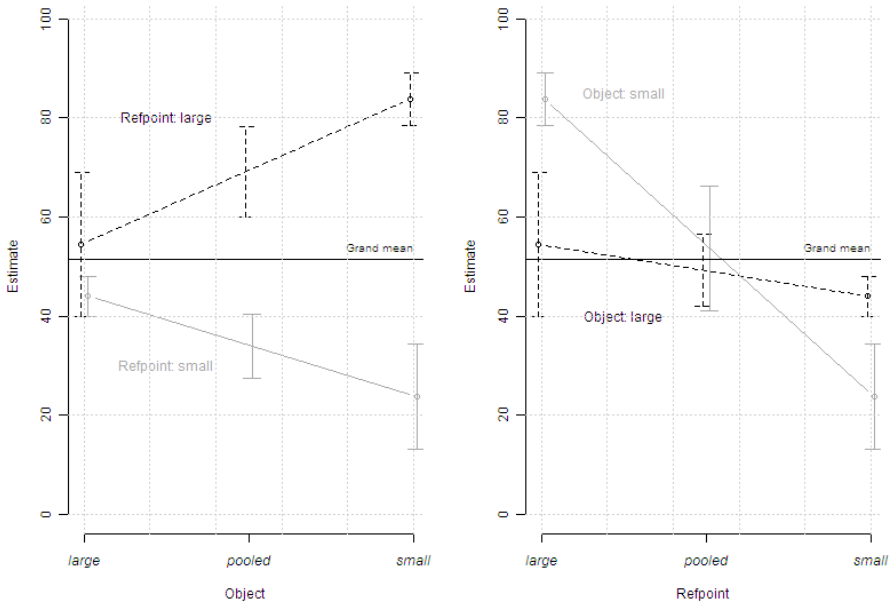


*Figure 66.* Interaction plot for ESTIMATE~OBJECT*REFPOINT

The graph gives away nearly everything one might want to know about the results, but you now compute the real statistical analysis: are the differences between the means significant or not? You begin by first testing the assumption of variance homogeneity:

```
> bartlett.test(ESTIMATE~OBJECT*REFPOINT)¶
........Bartlett·test·of·homogeneity·of·variances
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
data:··ESTIMATE·by·OBJECT·by·REFPOINT
Bartlett's·K-squared·=·1.9058,·df·=·1,·p-value·=·0.1674
```

This condition is met so you can proceed as planned. Again you first tell R how to contrast the means with other (sum contrasts), and then you compute the linear model and the ANOVA. Since you want to test both the two independent variables and their interaction you combine the two variables with an asterisk (cf. Section 3.2.2.2 above).

```
> options(contrasts=c("contr.sum",·"contr.poly"))¶
> model<-lm(ESTIMATE·~·OBJECT*REFPOINT)¶
> summary(model)¶
[…]
Coefficients:
·····················Estimate·Std.·Error·t·value·Pr(>|t|)
(Intercept)·········51.500·····4.796··10.738·1.65e-07
OBJECT1·············-2.250·····4.796··-0.469··0.64738
REFPOINT1···········17.625·····4.796···3.675··0.00318
OBJECT1:REFPOINT1··-12.375·····4.796··-2.580··0.02409
[…]
Residual·standard·error:·19.18·on·12·degrees·of·freedom
Multiple·R-Squared:·0.6294,·····Adjusted·R-squared:·0.5368
F-statistic:·6.794·on·3·and·12·DF,··p-value:·0.00627
```

With sum contrasts, in the first column of the coefficients, the first row (intercept) again provides the overall mean of the dependent variable: 51.5 is the mean of all estimates. The next two rows of the first column provide the differences for the alphabetically first factor levels of the respective variables: the mean of OBJECT: *LARGE* is 2.25 smaller than the overall mean and the mean of REFPOINT: *LARGE* is 17.625 larger than the overall mean. The third row of the first column shows how the mean of OBJECT: *LARGE* and REFPOINT: *LARGE* differs from the overall mean: it is 12.375 smaller. Again, the output of `model.tables` shows this in a little easier way:

```
> model.tables(aov(ESTIMATE·~·OBJECT*REFPOINT))¶
Tables·of·effects
·OBJECT
OBJECT
large·small
-2.25··2.25
·REFPOINT
REFPOINT
··large···small
·17.625·-17.625
·OBJECT:REFPOINT
·······REFPOINT
OBJECT··large···small
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
··large·-12.375··12.375
··small··12.375·-12.375
```

Back to `summary(model)`. The columns "Std. Error" and following provide the standard errors, *t*-values, and *p*-values. The coefficient of OBJECT is smaller than its standard error and not significant, but the coefficients for REFPOINT and the interaction of OBJECT and REFPOINT are significant. As for the whole model, there is a very significant correlation, given the *p*-value of 0.00627.

Let us now turn to the ANOVA table. For ANOVAs with more than one independent variable, you use the function `Anova` from the `library(car)` with the additional argument `type="III"`. (You can also use that function for monofactorial ANOVAs and would get the same results as discussed above, but for multifactorial ANOVAs you *must* use it to get the results you typically get with the default settings of other statistics software; cf. n. 45 again.)

```
> Anova(model,·type="III")¶
Anova·Table·(Type·III·tests)
Response:·ESTIMATE
···············Sum·Sq·Df··F·value····Pr(>F)
(Intercept)·····42436··1·115.3022·1.651e-07
OBJECT··········81··1···0.2201··0.647385
REFPOINT·······4970··1··13.5046··0.003179
OBJECT:REFPOINT···2450··1···6.6575··0.024088
Residuals·····4417·12
```

Of course the results do not change and the *F*-values are just the squared *t*-values.[48] The factor OBJECT is not significant, but REFPOINT and the interaction are. Unlike in Section 5.2, this time we do not compute a second analysis without OBJECT because even though OBJECT is not significant itself, it does participate in the significant interaction.

How can this result be interpreted? According to what we said about interactions above (e.g., Section 1.3.2.3) and according to Figure 66, REFPOINT is a very significant main effect – on the whole, large reference points increase the estimates – but this effect is actually dependent on the levels of OBJECT such that it is really only pronounced with small objects.

48. If your model included factors with more than two levels, the results from `Anova` would differ because you would get one *F*-value and one *p*-value for each predictor (factor / interaction) as a whole rather than a *t*-value and a *p*-value for one level (cf. also Section 5.4). Thus, it is best to look at the ANOVA table from `Anova(…,·type="III")`.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

For large objects, the difference between large and small reference points is negligible and, as we shall see in a bit, not significant.

To now also find out which of the independent variables has the largest effect size, you can apply the same logic as above and compute $\eta^2$. Obviously, REFPOINT has the strongest effect:

```
> 81/(81+4970+2450+4417)·#·for·OBJECT¶
[1]·0.006796442
> 4970/(81+4970+2450+4417)·#·for·REPFOINT¶
[1]·0.4170163
> 2450/(81+4970+2450+4417)·#·for·the·interaction¶
[1]·0.2055714
```

You can again perform pairwise *post-hoc* comparisons of the means. You find that, for the main effects, the conservative Tukey HSD test returns the same *p*-values as the ANOVA table (because for binary factors only one test is performed), and for the interaction, it returns only one significant difference and one that some people call marginally significant.

```
> TukeyHSD(aov(model),·ordered·=·T)¶
··Tukey·multiple·comparisons·of·means
····95%·family-wise·confidence·level
····factor·levels·have·been·ordered
Fit:·aov(formula·=·model)
$OBJECT
·····diff·····lwr·····upr····p·adj
small-large··4.5·-16.39962·25.39962·0.6473847

$REPFOINT
·····diff·····lwr·····upr····p·adj
large-small·35.25·14.35038·56.14962·0.0031786

$`OBJECT:REFPOINT`
·····diff·····lwr·····upr····p·adj
large:small-small:small·20.25·-20.024414··60.52441·0.4710945
large:large-small:small·30.75··-9.524414··71.02441·0.1607472
small:large-small:small·60.00··19.725586·100.27441·0.0039895
large:large-large:small·10.50·-29.774414··50.77441·0.8646552
small:large-large:small·39.75··-0.524414··80.02441·0.0534462
small:large-large:large·29.25·-11.024414··69.52441·0.1908416
```

Let us now turn to the diagnostic plots of the linear model.

```
> par(mfrow=c(2,·2))¶
> plot(model)¶
> par(mfrow=c(1,·1))·#·restore·the·standard·plotting·setting¶
```

You can see that the results are not as good as one would have hoped

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

for (which is of course partially due to the fact that this is a very small and invented data set). The Bartlett test above yielded the desired result, and the plot on the top left shows that residuals are nicely distributed around 0, but both plots on the left are not completely unstructured. In addition, the upper right plot reveals some deviation from normality, which however turns out to be not significant.

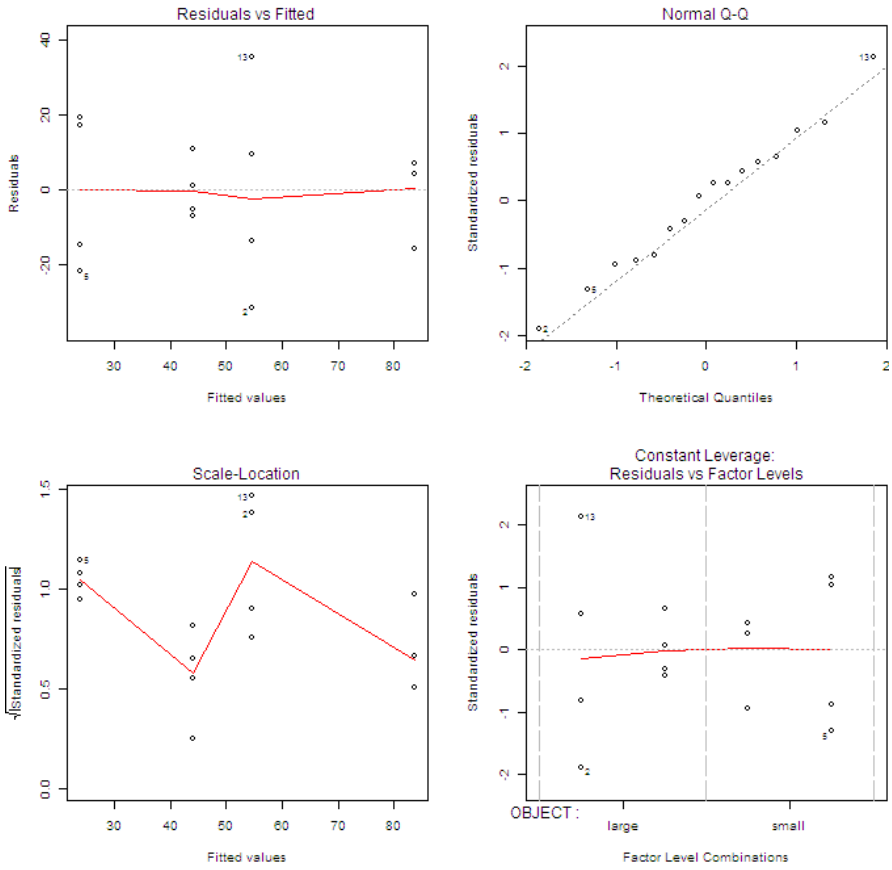


*Figure 67.* Model-diagnostic plots

```
> shapiro.test(residuals(model))¶
        Shapiro-Wilk normality test
data:  residuals(model)
W = 0.9878, p-value = 0.9973
```

The plot on the bottom right shows some differences between the resi-

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

duals both in the whole sample and between the four groups in the sample. However, for expository reasons and since the data set is too small and invented, and because `library(gvlma);·gvlma(model)¶` does not return significant violations, let us for now just summarize the results.

"There is an intermediately strong correlation between the size of the average estimate and the sizes of the quantified objects and their reference points; according to a two-factor ANOVA, this correlation is very significant ($F_{3, 12} = 6.79$; $p < 0.0063$ **) and explains more than 50% of the overall variance (multiple adj. $R^2 = 0.537$). However, the size of the quantified object alone does not contribute significantly: $F_{1, 12} < 1$; $p = 0.647$; $\eta^2 < 0.01$. The size of the reference point has a very significant influence on the estimate ($F_{1, 12} = 13.5$; $p < 0.0032$; $\eta^2 = 0.417$) and the strongest effect size ($\eta^2 = 0.417$). The direction of this effect is that large and small reference points yield larger and smaller estimates respectively. However, there is also a significant interaction showing that large reference points really yield large estimates with small objects only ($F_{1, 12} = 6.66$; $p = 0.024$; $\eta^2 = 0.206$). [Insert Figure 66 and the tables resulting from `model.tables(aov(ESTIMATE~OBJECT* REFPOINT),·"means")¶`.] Pairwise post-hoc comparisons with Tukey's HSD tests show the most reliable differences between means arise between OBJECT: *small* / REFPOINT: *LARGE* and OBJECT: *SMALL* / REFPOINT: *SMALL*."

**Recommendation(s) for further study**

- the function `adonis` from the `library(vegan)` to compute a multivariate non-parametric alternative to ANOVAs
- the function `all.effects` and others from the `library(effects)` to represent effects in linear models graphically
- the function `rpart` from the `library(rpart)`, to compute classification and regression trees as an alternative to linear models (cf. the exercise files for examples)
- Backhaus et al. (2003: Ch. 2), Crawley (2002: Ch. 13, 15, 17), Crawley 2005: Ch. 7, 9, 12), Crawley (2007: Ch. 11, 12), Faraway (2005: Ch. 14), Baayen (2008: Section 4.4, Ch. 6-7), Johnson (2008: Ch. 4)

## 4. Binary logistic regression

In the last two sections, we dealt with methods in which the dependent variable is ratio-scaled. However, in many situations the dependent variable

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

is binary/nominal (or categorical, which we are not going to be concerned with). In such situations, one should not use a linear model of the kind discussed so far (with a so-called identity link) because the values of the dependent variable can only take on two values (which should be coded as 0 and 1 or, preferably, as a two-level factor, as I mentioned above in Section 1.3.2.2, note 3) while the values predicted from the above kind of linear model will be continuous and theoretically unbound. In addition, one should also not compute a linear model in which the dependent variable corresponds to the percentages of the levels of the 0/1-coded nominal variable because a linear model of the above kind will predict negative values, i.e., values outside of the meaningful range for percentages (cf. above note 42). Even some of the traditionally used transformations (such as the arcsine transformation) that have been applied to nominal dependent variables can be problematic (cf. Jaeger 2008). Thus, for binary dependent variables, binary logistic regression is often the best choice.

As an example, we consider another instance of a constructional alternation, this time the so-called dative alternation in English exemplified in (54) and (55). (By the way, it seems to be nearly a running gag that every introduction to statistics for linguists with R has at least one example involving the dative alternation. I did not dare break that tradition …)

| | | | |
|---|---|---|---|
| (54) | a. | He gave his father a book. | (ditransitive) |
| | b. | He gave a book to his father. | (prep. dative$_{to}$) |
| (55) | a. | He made his father a sandwich. | (ditransitive) |
| | b. | He made a sandwich for his father. | (prep. dative$_{for}$) |

We concentrate only on the alternation in (54): ditransitives vs. prepositional datives with *to*. While a study of the literature would again reveal a large number of variables influencing speakers' choices of constructions, we focus on the following:

- one dependent nominal variable, namely the choice of construction: CONSTRUCTION: *0/DITRANSITIVE* for the ditransitive and CONSTRUCTION: *1/PREP_DATIVE* for the prepositional dative with *to*;
- one independent nominal variable, namely V_CHANGPOSS: *NO_CHANGE* (for examples such as *He gave me a hard time*, which do not imply literal change of possession by transfer of an entity) and V_CHANGPOSS: *CHANGE* (for examples such as *He gave me a book*, which do);
- three independent ratio-scaled variables, namely AGENT_ACT, REC_ACT, and PAT_ACT, whose values indicate how familiar, or active,

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

the referent of the agent, the recipient, and the patient of the relevant clause are from the preceding context: high values reflect high familiarity because, say, the referent has been mentioned before often.

A binary logistic regression involves the following procedure:

| **Procedure** |
|---|
| Formulating the hypotheses |
| Inspecting graphical representations (plus maybe computing some descriptive statistics) |
| Testing the assumption(s) of the test: checking for overdispersion |
| Computing the multiple correlation Nagelkerke's $R^2$ and differences of probabilities (and maybe odds) |
| Computing the test statistic likelihood ratio chi-square, the degrees of freedom *df*, and the probability of error *p* |

First, the hypotheses:

$H_0$: The frequencies of the two constructions are independent of the independent variables and their pairwise interactions: $R^2 = 0$.

$H_1$: The frequencies of the two constructions are not independent of the independent variables and their pairwise interactions: $R^2 > 0$.

You clear the memory and load the data from <C:/_sflwr/_inputfiles/05-4_dativealternation.txt>:

```
> rm(list=ls(all=T))¶
> DatAlt<-read.table(choose.files(),·header=T,·sep="\t",·
      comment.char="",·quote="")¶
> summary(DatAlt)
·····CASE·····CONSTRUCTION1·····CONSTRUCTION2
·Min.···:··1.0···Min.···:0.0···ditransitive:200··
·1st·Qu.:100.8···1st·Qu.:0.0···prep_dative·:200··
·Median·:200.5···Median·:0.5·····················
·Mean···:200.5···Mean···:0.5·····················
·3rd·Qu.:300.2···3rd·Qu.:1.0·····················
·Max.···:400.0···Max.···:1.0·····················
····V_CHANGPOSS····AGENT_ACT·····REC_ACT·····PAT_ACT···
·change···:252···Min.···:0.00···Min.···:0.00···Min.···:0.000
·no_change:146···1st·Qu.:2.00···1st·Qu.:2.00···1st·Qu.:2.000
·NA's·····2···Median·:4.00···Median·:5.00···Median·:4.000
···············Mean···:4.38···Mean···:4.63···Mean···:4.407
···············3rd·Qu.:7.00···3rd·Qu.:7.00···3rd·Qu.:7.000
···············Max.···:9.00···Max.···:9.00···Max.···:9.000
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

Note first that the dependent variable is (redundantly) included twice, once as a vector of zeroes and ones (CONSTRUCTION1) and once as a factor (CONSTRUCTION2) with the constructions as its levels. Second, there are two cases with missing data, which you can eliminate by instructing R to retain only the rows with complete cases. If you check the structure of the new version of `DatAlt`, you see that only the 398 complete cases are used.

```
> DatAlt<-DatAlt[complete.cases(DatAlt),]¶
> attach(DatAlt)¶
```

You begin by looking at the observed frequencies in a purely descriptive way using graphs: association or mosaic plots for the categorical variable V_CHANGPOSS and spineplots (and/or conditional density plots using `cdplot`) for the three other independent variables.

```
> assocplot(table(V_CHANGPOSS·,·CONSTRUCTION2)¶
> spineplot(CONSTRUCTION2·~·AGENT_ACT)¶
> spineplot(CONSTRUCTION2·~·REC_ACT)¶
> spineplot(CONSTRUCTION2·~·PAT_ACT)¶
```

The graphs suggest that V_CHANGPOSS and CONSTRUCTION interact: when the clause implies a change of possession, then the ditransitive construction is preferred, but when the clause does not, ditransitives are dispreferred. AGENT_ACT does not seem to have a particularly systematic influence on the choice of construction, whereas PAT_ACT and REC_ACT appear to have clear effects. For a more comprehensive analysis, you of course now need to check the effect(s) of all variables at the same time. For here, we assume you want to test (i) whether each variable has a significant effect on the choice of construction in isolation (i.e., whether what you see in the graphs is significant) and (ii) whether the interactions of each ratio-scaled independent variable with V_CHANGPOSS are significant.

Basically, you go through a similar model selection process as in Section 5.2. You begin with the most complete model you are interested in and successively remove the interactions and (later) variables with the highest *p*-values, always checking with `anova` whether your simplifications are justified. Two differences are important: first, you use not `lm`, but the function `glm` (for *generalized linear model*). Just like `lm`, `glm` is used with a formula and, for now, the standard contrast settings, but we also need the additional argument `family=binomial`, which tells R that you want a binary logistic regression (rather than the 'normal' linear regression from above), which in turn makes sure that the values predicted by the model fall

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

into the range between 0 and 1.[49] Second, the anova test is done with the additional argument test="Chi".

```
> options(contrasts=c("contr.treatment", "contr.poly"))¶
> model.glm<-glm(CONSTRUCTION1~V_CHANGPOSS*AGENT_ACT+
      V_CHANGPOSS*REC_ACT+V_CHANGPOSS*PAT_ACT,·
      family=binomial)¶
>·summary(model.glm)¶
[…]
Deviance·Residuals:
····Min·····1Q···Median·····3Q·····Max
-2.3148··-0.7524··-0.2891···0.7716···2.2402

Coefficients:
·····························Estimate·Std.·Error·z·value·Pr(>|z|)
(Intercept)·····················-1.131682···0.430255··-2.630··0.00853
V_CHANGPOSSno_change···········1.110805···0.779622···1.425··0.15422
AGENT_ACT·····················-0.022317···0.051632··-0.432··0.66557
REC_ACT·····················-0.179827···0.054679··-3.289··0.00101
PAT_ACT·····················0.414957···0.057154···7.260·3.86e-13
V_CHANGPOSSno_change:AGENT_ACT··0.001468···0.094597···0.016··0.98761
V_CHANGPOSSno_change:REC_ACT···-0.164878···0.102297··-1.612··0.10701
V_CHANGPOSSno_change:PAT_ACT····0.062054···0.105492···0.588··0.55637
[…]
(Dispersion·parameter·for·binomial·family·taken·to·be·1)
····Null·deviance:·551.74··on·397··degrees·of·freedom
Residual·deviance:·388.64··on·390··degrees·of·freedom
AIC:·404.64
Number·of·Fisher·Scoring·iterations:·5
```

The output is similar to what you know from lm. I will say something about the two lines involving the notion of deviance below, but for now you can just proceed with the model selection process as before:

```
> model.glm.2<-update(model.glm,·~.·-·V_CHANGPOSS:AGENT_ACT)¶
> summary(model.glm.2);·anova(model.glm,·model.glm.2,·
      test="Chi")¶
> model.glm.3<-update(model.glm.2,·~.·-·V_CHANGPOSS:PAT_ACT)¶
> summary(model.glm.3);·anova(model.glm.2,·model.glm.3,·
      test="Chi")¶
> model.glm.4<-update(model.glm.3,·~.·-·V_CHANGPOSS:REC_ACT)¶
> summary(model.glm.4);·anova(model.glm.3,·model.glm.4,·
      test="Chi")¶
> model.glm.5<-update(model.glm.4,·~.·-·AGENT_ACT)¶
> anova(model.glm.4,·model.glm.5,·test="Chi");·
      summary(model.glm.5)¶
[…]
```

49. I simplify here a lot and recommend Crawley (2007: Ch. 13) for more discussion of link functions.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
·····················Estimate·Std.·Error·z·value·Pr(>|z|)
(Intercept)··········-1.07698····0.32527···-3.311··0.00093
V_CHANGPOSSno_change··0.61102····0.26012···2.349··0.01882
REC_ACT··············-0.23466····0.04579···-5.125·2.98e-07
PAT_ACT··············0.43724····0.04813···9.084··<·2e-16

(Dispersion·parameter·for·binomial·family·taken·to·be·1)
····Null·deviance:·551.74··on·397··degrees·of·freedom
Residual·deviance:·391.72··on·394··degrees·of·freedom
AIC:·399.72
```

That's the minimal adequate model: all remaining variables are significant. To get a nice summary of the results that involves an $R^2$-value and some other useful statistics, I advise you to use another very useful function, `lrm` (from the `library(Design)`), which requires the already familiar kind of formula. For using the library `Design`, it is useful to also define a data distribution object which internally summarizes the distribution of your data.

```
> library(Design)¶
> DatAlt.dd<-datadist(DatAlt);·options(datadist="DatAlt.dd")¶
```

Then you generate a model object (with either the vector `CONSTRUCTION1` or the factor `CONSTRUCTION2` as dependent variable) and look at the results (I omit the call in the results output):

```
> model.lrm<-lrm(CONSTRUCTION1·~·V_CHANGPOSS·+·REC_ACT·+·
       PAT_ACT,·x=T,·y=T,·linear.predictors=T);·model.lrm¶
[…]
Frequencies·of·Responses
··0···1
200·198
··Obs··Max·Deriv·Model·L.R.·······d.f.··········P·······C
··398·····3e-10·····160.01·········3·········0···0.842
·····Dxy·····Gamma·····Tau-a·····R2·····Brier
·····0.685·····0.688·····0.343·····0.441····· 0.161

·····················Coef····S.E.····Wald·Z·P
Intercept·····-1.0770·0.32528·-3.31··0.0009
V_CHANGPOSS=no_change··0.6110·0.26012··2.35··0.0188
REC_ACT··············-0.2347·0.04579·-5.12··0.0000
PAT_ACT··············0.4372·0.04814··9.08··0.0000
```

The coefficients at the bottom are the same, but now you also get some more information. There is a highly significant correlation between the remaining three variables and the choice of construction: the model's likelihood ratio chi-square is 160.01 (the difference between the two deviance

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

values from the `glm` output) with $df = 3$ (the difference between the $df$-values of the two deviance values from the `glm` output) and $p \approx 0$. Then, there is a variety of classification accuracy measures $C$, $D_{xy}$, etc. These measures answer the question of how good the model is at classifying the chosen construction for each analyzed instance. $C$ is a coefficient of concordance, which can be considered good when it reaches or exceeds approximately 0.8, which it does here. Somer's $D_{xy}$ is a rank correlation between the predicted probabilities of the two constructions and the actually observed constructions. Its value falls between 0 and 1 and since it follows directly from $C$ it is also good. Gamma, $tau_a$, and Brier are comparable coefficients I will not discuss here, and the $R^2$-value (here it is called Nagelkerke's $R^2$) and its general meaning as an indicator of correlational strength you already know.

As for the sizes of the effects, the more a coefficient in the above `glm` or `lrm` output deviates from 0, the stronger – on the whole – the observed effect. Why is that and what do the coefficients mean anyway? Put differently, what do they try to predict? To understand that, we look at columns of two 'randomly' chosen cases:

```
> DatAlt[c(1, 300), -c(1, 2, 5)]¶
    CONSTRUCTION2 V_CHANGPOSS REC_ACT PAT_ACT
1    ditransitive    no_change       8       0
313  prep_dative       change       3       8
```

That means, for these two cases the regression equation returns the following predictions:

```
> c1<-sum(-1.0770, 1*0.6110, 8*-0.2347, 0*0.4372); c1¶
 [1] -2.3436
> c313<-sum(-1.0770, 0*0.6110, 3*-0.2347, 8*0.4372); c313¶
 [1] 1.7165
```

These are obviously neither the values 0 or 1 and not even just values between 0 and 1 – these are values between $-\infty$ and $+\infty$. The values are so-called *logits*, i.e., the logarithms of the odds of the event to be predicted, which is here – and this is very important! – the variable level coded with 1 or, in the case of a factor, the second level, i.e., here the prepositional dative (recall from Section 4.1.2.2: the odds of an event $E$ are defined as $p_E/(1-p_E)$). However, you can compute what are called the *inverse logits*, which lie between 0 and 1 and correspond to the probabilities of the predicted construction. These inverse logits are computed as follows:

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> 1/(1+exp(-c1))·#·or·exp(c1)/(1+exp(c1))¶
[1]·0.087576
> 1/(1+exp(-c1))·#·or·exp(c1)/(1+exp(c1))¶
[1]·0.84768
```

For case #1, the model predicts only a probability of 8.7576% of a prepositional dative, given the behavior of the three independent variables (which is good, because our data show that this was in fact a ditransitive). For case #313, the model predicts a probability of 84.768% of a prepositional dative (which is also good, because our data show that this was indeed a prepositional dative). These two examples are obviously cases of firm and correct predictions, and the more the coefficient of a variable deviates from 0, the more this variable can potentially make the result of the regression equation deviate from 0, too, and the larger the predictive power of this variable. Look what happens when you compute the inverse logit of 0: the probability becomes 50%, which with two alternatives is of course the probability with the lowest predictive power.

```
> 1/(1+exp(0))¶
[1]·0.5
```

Let's return to the coefficients now. In this case, V_CHANGPOSS has a strong effect and since its coefficient is positive, on the whole, when V_CHANGPOSS is *NO_CHANGE*, then this *in*creases the probability for CONSTRUCTION: *1* (the prepositional dative). For example, the above case of `c313` is a data point with V_CHANGPOSS:*CHANGE*. If you changed case 313 into a hypothetical case 313' with V_CHANGPOSS:*NO_CHANGE* and left everything else as is, then the predicted probability of the prepositional dative would increase from 84.768% to 91.113% (i.e., by 6.345%):

```
> c313.hyp<-sum(-1.0770,·1*0.6110,·3*-0.2347,·8*0.4372);·
      c313.hyp¶
[1]·2.3275
> 1/(1+exp(-c313.hyp))¶
[1]·0.91113
```

Note: this does *not* mean that V_CHANGPOSS:*NO_CHANGE* uniformly increases the probability of CONSTRUCTION: *1* by 6.345%. How much does changing case 1 to a hypothetical case 1' with V_CHANGPOSS:*NO_CHANGE* influence the probability of CONSTRUCTION: *1*?

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

**THINK**
**BREAK**

```
> c1.hyp<-sum(-1.0770,·0*0.6110,·8*-0.2347,·0*0.4372);·
      c1.hyp¶
[1]·2.9546
> 1/(1+exp(-c1.hyp))¶
 [1]·0.04952
```

The difference is only one of 3.8056%: the effects of the independent variables on the probability of the predicted prepositional dative are not linear on the probability scale – they are linear on the logit scale as you see when you compare `c1-c1.hyp¶` to `c313-c313.hyp¶`.

Back to the other variables … As for PAT_ACT, high values of PAT_ACT *in*crease the probability of CONSTRUCTION: *1*. On the other hand, REC_ACT has a negative coefficient, which is why high values of REC_ACT *de*crease the probability of CONSTRUCTION: *1*. Lastly, note that you can obtain confidence intervals for the coefficients by entering `confint(model.glm.5)¶`.

Sometimes you will find not the regression coefficients but the exponentiated regression coefficients that tell you how each variable influences the odds of the predicted outcome, the prepositional dative.

```
> exp(model.glm.5$coefficients)¶
·········(Intercept)·V_CHANGPOSSno_change
··········0.3406224···········1.8423187
···········REC_ACT···········PAT_ACT
··········0.7908433···········1.5484279

> exp(confint(model.glm.5))¶
Waiting·for·profiling·to·be·done...
·····················2.5·%···97.5·%
(Intercept)·········0.1774275·0.6372445
V_CHANGPOSSno_change·1.1100411·3.0844025
REC_ACT·············0.7213651·0.8635779
PAT_ACT·············1.4135976·1.7079128
```

The overall logic of the interpretation of the directionality etc. remains the same. If PAT_ACT increases by one unit, this increases the odds of prepositional datives – the ratio $p_{\text{prep.dat.}}/p_{\text{ditrans.}}$ – by about 1.55. One nice way of inspecting all main effects is available by plotting the `lrm` model object; the argument `fun=plogis` instructs R to have the predicted probabilities (rather than log odds) on the *y*-axes.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> par(mfrow=c(1,·3))¶
> plot(model.lrm.2,·fun=plogis,·ylim=c(0,·1),·
      ylab="Probability·of·prepositional·dative")¶
> par(mfrow=c(1,·1))·#·restore·the·standard·plotting·setting¶
```

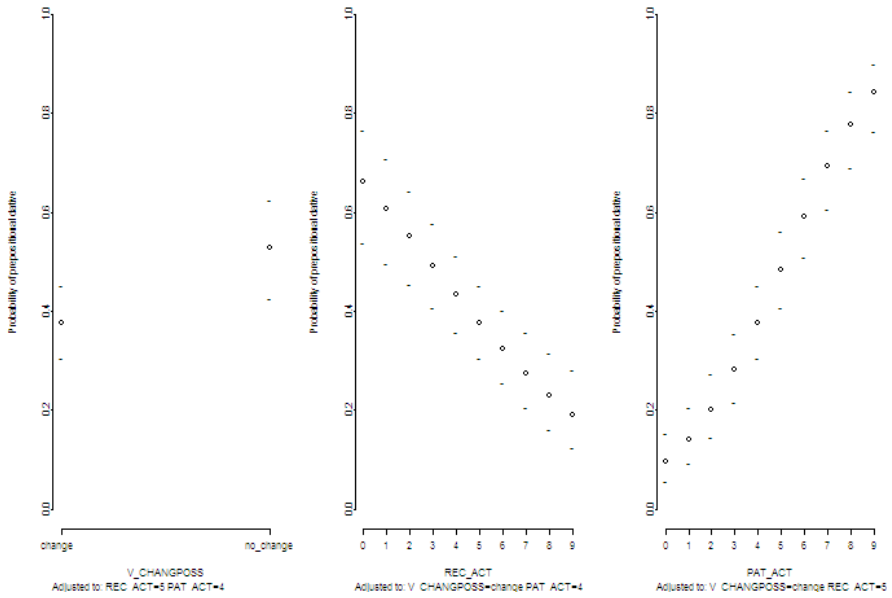


*Figure 68.* Partial effects for `model.lrm`

This basically confirms the results of the monofactorial exploration of the graphs at the beginning of this section, which is not really a big surprise given that all interactions were filtered out during the model selection process. Note, however, that the effects plotted here are not averages across the whole data set, but each variable's effect shown is computed with the other variables set to a particular level (for the ratio-scaled variable, that is the median); cf. the subtitles! If a variable participates in an interaction, it can therefore happen that the plot's result is at odds (no pun intended) with the overall numerical result because the plotted effect is only true for the variable level(s) for which the trend is computed, but not for others or in general.

Instead of the above kind of model summary, you will also sometimes find ANOVA tables reported for such analyses. There are two ways of getting these, which here provide identical results. One is applying `anova`

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

to `model.lrm`, the other again requires the function `Anova` from the `library(car)` with sum contrasts (!); I only show the output of the latter:

```
> library(car)·#·for·the·function·Anova¶
> options(contrasts=c("contr.sum",·"contr.poly"))¶
> Anova(model.glm.5,·type="III",·test.statistic="Wald")¶
Anova·Table·(Type·III·tests)
Response:·CONSTRUCTION1
·············Wald·Chisq·Df·Pr(>Chisq)
(Intercept)·····10.963··1··0.0009296·***
V_CHANGPOSS·····5.518··1··0.0188213·*
REC_ACT·········26.264··1··2.977e-07·***
PAT_ACT·········82.517··1··<·2.2e-16·***
> options(contrasts=c("contr.treatment",·"contr.poly"))¶
```

Here, where all variables are binary factors or ratio-scaled, we get the same *p*-values, but with categorical factors, the ANOVA output provides one *p*-value for the whole factor (recall note 48 above). The Wald chi-square values correspond to the squared *z*-value of the `glm` output.

Let us now briefly return to the interaction between REC_ACT and V_CHANGPOSS. This interaction was deleted during the model selection process, but I would still like to discuss it briefly for three reasons. First, compared to many other deleted predictors, its *p*-value was still rather small (around 0.12); second, an automatic model selection process with `step` actually leaves it in the final model (although an `anova` test would encourage you to delete it anyway); third, I would briefly like to mention how you can look at such interactions (in a simple, graphical manner similar to that used for multiple regression). Let's first look at the simple main effect of V_CHANGPOSS in terms of the percentages of constructions:

```
> prop.table(table(CONSTRUCTION2,·V_CHANGPOSS),·2)¶
··············V_CHANGPOSS
CONSTRUCTION2·····change·no_change
··ditransitive·0.5555556·0.4109589
··prep_dative··0.4444444·0.5890411
```

On the whole – disregarding other variables, that is – this again tells you V_CHANGEPOSS:*CHANGE* increases the likelihood of ditransitives. But you know that already and we now want to look at an interaction. As above, one quick and dirty way to explore the interaction consists of splitting up the ratio-scaled variable REC_ACT into groups esp. since its different values are all rather equally frequent. Let us try this out and split up REC_ACT into two groups (which can be done on the basis of the mean or the median, which are here very close to each other anyway):

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> REC_ACT.larger.than.its.mean<-REC_ACT>mean(REC_ACT)¶
```

(Note that by its very nature such dichotomization loses information, which is why you must be careful and not apply this prematurely; I will mention below how you check the interaction without dichotomization, too.) Then, you can generate a table that represents the relevant interaction:

```
> interact.table<-table(CONSTRUCTION2,·REC_ACT.larger.
      than.its.mean,·V_CHANGPOSS);·interact.table¶

,·,·V_CHANGPOSS·=·change
·····················REC_ACT.larger.than.its.mean
CONSTRUCTION2·····FALSE·TRUE
··ditransitive·····51···89
··prepositional·dative····67···45

,·,·V_CHANGPOSS·=·no_change
·····················REC_ACT.larger.than.its.mean
CONSTRUCTION2·····FALSE·TRUE
··ditransitive·····18···42
··prepositional·dative····54···32
```

You get one table with two subtables: one for when change is implied and one for where it is not. But ideally, we of course look at the constructional percentages – not just their absolute values. You therefore generate percentage tables for the two subtables of both levels of V_CHANGPOSS (with `prop.table` and column percentages). Note how you use subsetting with two commas (!) to access the first and the second sub-table: `[,,1]` and `[,,2]`.

```
> interact.table.1.perc<-prop.table(interact.table[,,1],
      ·2)·#·note·how·you·get·the·first·subtable:·[,,1]¶
> interact.table.2.perc<-prop.table(interact.table[,,2],·
      2)·#·note·how·you·get·the·first·subtable:·[,,1]¶
```

Then you can simply do two bar plots. I only show the simplest possible code here, but a more customized graph whose code you find in the code file):

```
> barplot(interact.table.1.perc,·beside=T,·legend=T)¶
> barplot(interact.table.2.perc,·beside=T,·legend=T)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

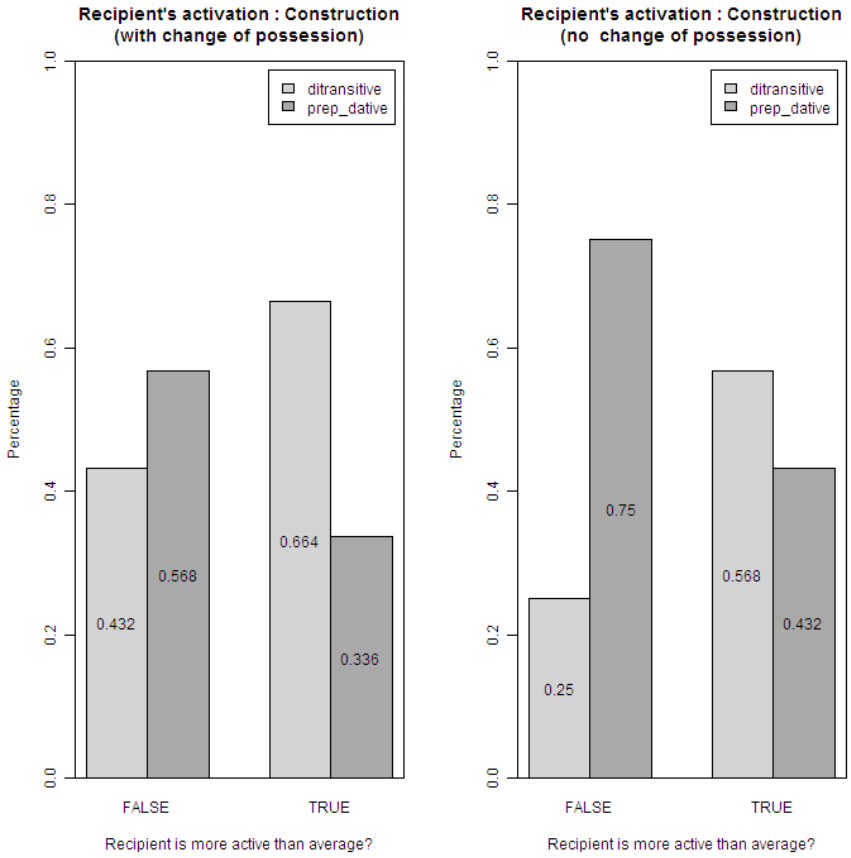


*Figure 69.* Bar plots for CONSTRUCTION1~V_CHANGPOSS*PAT_ACT

The graphs show what the (insignificant) interaction was about: when no change is implied (i.e., in the right panel), then the preference of rather unfamiliar recipients (i.e., cf. the left two bars) for the prepositional dative is somewhat stronger (0.75) than when change is implied (0.568; cf. the two left bars in the left panel). But when no change is implied (i.e., in the right panel), then the preference of rather familiar recipients (i.e., cf. the right two bars) for the prepositional dative is somewhat stronger (0.432) than when change is implied (0.336). But then, this effect did not survive the model selection process … (Cf. the code file for what happens when you split up REC_ACT into three groups – after all, there is no particularly good reason to assume only two groups – as well as further down in the

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

code file, Gelman and Hill 2007: Section 5.4, or Everitt and Hothorn 2006: 101–3 for how to plot interactions without grouping ratio-scaled variables.)

Let us now turn to the assessment of the classification accuracy. The classification scores in the `lrm` output already gave you an idea that the fit of the model is pretty good, but it would be nice to assess that a little more straightforwardly. One way to assess the classification accuracy is by comparing the observed constructions to the predicted ones. Thankfully, you don't have to compute every case's constructional prediction like above. You have seen above in Section 5.2 that the function `predict` returns predicted values for linear models, and the good news is that `predict` can also be applied to objects produced by `glm` (or `lrm`). If you apply `predict` to the object `model.glm.5` without further arguments, you get the values following from the regression equation (with slight differences due to rounding):

```
> predict(model.glm.5)[c(1, 313)]¶
     1     313
-2.3432  1.7170
```

However, we want easily interpretable probabilities. Thus, you add the argument `type="response"`:

```
> predict(model.glm.5, type="response")[c(1, 313)]¶
       1       313
0.087608 0.847739
```

The easiest way is to generate a vector `classifications`, which contains for each instance the predicted probability of the prepositional dative.

```
> classifications<-predict(model.glm.2, type="response")¶
```

(You could also generate `classifications` with `model.glm.2$fitted¶` or `predict(model.lrm.2, type="fitted")¶`.) Then, you can easily convert this into a categorical decision: since there are only two constructional possibilities, you can simply recode `classifications` into `classifications.2` with "ditransitive" when the probability of a prepositional dative is smaller than 0.5 and as "prep_dative" when the probability of a prepositional dative is 0.5 or greater. Then, all you need to do is crosstabulate these classifications with the actually observed constructions:

```
> classifications.2<-ifelse(classifications>=0.5,
      "prep_dative","ditransitive")¶
> evaluation<-table(classifications.2, CONSTRUCTION2)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> addmargins(evaluation)¶
·····················CONSTRUCTION2
classifications.2·ditransitive·prep_dative·Sum
··ditransitive·············155·········46·201
··prep_dative··············45··········152·197
··Sum·····················200·········198·398
```

The correct predictions are in the main diagonal of this table: 155+152 = 307 of the 398 constructional choices are correct, which corresponds to 77.14%:

```
> sum(diag(evaluation))/sum(evaluation)¶
[1]·0.7713568
```

It's now also possible to look at the characteristics of the misclassified constructions. I leave that up to you for now, but we will return to this issue in the exercises.

Let us finally briefly look at model assumptions. Compared to, say, linear discriminant analysis, another method that can be used to predict nominal/categorical variables, binary logistic regression does not come with very many distributional assumptions, and many textbooks do not discuss any at all (but cf. Harrell 2001 and Faraway 2006: Section 6.4). For now, we just assume that the data points are independent of each other (because, for example, every constructional choice is from a different corpus file).

The criterion of overdispersion requires that you look at the ratio of the residual deviance and the residual *df*s in the `glm` output. The ratio of the two should not be much larger than 1. In the present case, the ratio is $^{323.83}/_{295} \approx 1.1$, which is a pretty good result. Several references just say that, if you get a value that is much larger than 1, e.g. > 2, then you would run the `glm` analysis again with the argument `family=quasibinomial` and then take it from there. Baayen (2008: 199), quoting Harrell (2001: 231), recommends as a heuristic a chi-square test of the residual deviance at the residual *df*, which here returns the desired insignificant result:

```
> pchisq(391.72,·394,·lower.tail=F)¶
[1]·0.5229725
```

To sum up: "A binary logistic regression shows that there is a highly significant intermediately strong correlation between some of the independent variables and the constructional choice: Log-likelihood ratio $\chi^2$ = 160.01; $df$ = 1; $p < 0.001$. Nagelkerke's $R^2$ is 0.441, and the minimal adequate model has a good classificatory power: $C$ = 0.842, $D_{xy}$ = 0.685, and

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

77.14% of the constructions are predicted correctly. The variable V_CHANGPOSS has a strong effect (odds ratio ≈ 1.84; 95%-CI: 1.11 and 3.08; $p = 0.0188$): change of possession prefers the ditransitive constructions. The variable PAT_ACT also exhibits a significant correlation with the constructional choice (odds ratio ≈ 1.55; 95%-CI: 1.41 and 1.71; $p < 0.001$): the more the patient is known/given from the discourse context, the more the prepositional dative is preferred. The final significant effect is that of REC_ACT: the more the recipient is known/given, the more the ditransitive is preferred (odds ratio ≈ 0.79; 95%-CI: 0.72 and 0.87; $p < 0.0001$); both latter effects are compatible with information-structural *given-before-new* analyses. [plus illustrate the effects with some graphs; cf. the many examples in the code file]"

**Recommendation(s) for further study**

- just like in Section 5.2, it can also help interpreting the regression coefficients when the input variables are centered
- the function `polr` from the `library(MASS)`, to compute logistic regressions with ordinal factors as dependent variable (`lrm` can do that too, though)
- the function `rpart` from the `library(rpart)` to compute classification and regression trees as an alternative to logistic regressions
- the function `validate` from the `library(Design)` to do a resampling validation of a logistic regression computed with `lrm`
- Pampel (2000), Jaccard (2001), Crawley (2002: Ch. 30), Crawley 2005: Ch. 16), Crawley (2007: Ch. 17), Faraway (2006: Ch. 2, 6), Maindonald and Braun (2003: Ch. 8), Gelman and Hill (2007: Ch. 5), Baayen (2008: Section 6.3), Johnson (2008: Section 5.4)

## 5. Hierarchical agglomerative cluster analysis

With the exception of maybe the HCFA we have so far only concerned ourselves with methods in which independent and dependent variables were clearly separated and where we already had at least an expectation and a hypothesis prior to the data collection. Such methods are sometimes referred to as hypothesis-testing statistics, and we used statistics and *p*-values to decide whether or not to reject a null hypothesis. The method called hierarchical agglomerative cluster analysis that we deal with in this section is a so-called *exploratory*, or *hypothesis-generating*, method or, more precisely, a family of methods. It is normally used to divide a set of

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

elements into clusters, or groups, such that the members of one group are very similar to each other and at the same time very dissimilar to members of other groups. An obvious reason to use cluster analyses to this end is that this method can handle larger amounts of data and be at the same time more objective than humans eyeballing huge tables.

To get a first understanding of what cluster analyses do, let us look at a fictitious example of a cluster analysis based on similarity judgments of English consonant phonemes. Let's assume you wanted to determine how English native speakers distinguish the following consonant phonemes: /b/, /d/, /f/, /g/, /l/, /m/, /n/, /p/, /s/, /t/, and /v/. You asked 20 subjects to rate the similarities of all $^{(11 \cdot 10)}/_{2} = 55$ pairs of consonants on a scale from 0 ('completely different') to 1 ('completely identical'). As a result, you obtained 20 similarity ratings for each pair and could compute an average rating for each pair. It would now be possible to compute a cluster analysis on the basis of these average similarity judgments to determine (i) which consonants and consonant groups the subjects appear to distinguish and (ii) how these groups can perhaps be explained. Figure 70 shows the result that such a cluster analysis might produce - how would you interpret it?

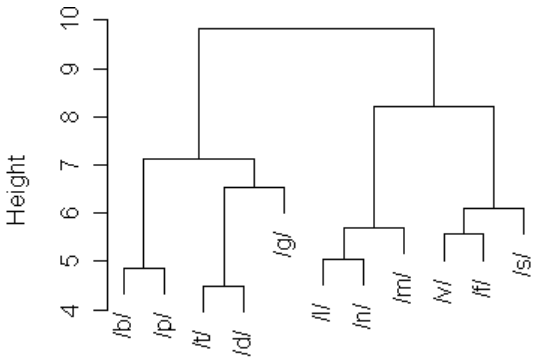


*Figure 70.* Fictitious results of a cluster analysis of English consonants

**THINK**
**BREAK**

The 'result' suggests that the subjects' judgments were probably strongly influenced by the consonants' manner of articulation: on a very general level, there are two clusters, one with /b/, /p/, /t/, /d/, and /g/, and one with

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

/l/, /n/, /m/, /v/, /f/, and /s/. It is immediately obvious that the first cluster contains all and only all plosives (i.e., consonants whose production involves a momentary *complete* obstruction of the airflow) that were included whereas the second cluster contains all and only all nasals, liquids, and fricatives (i.e., consonants whose production involves only a momentary *partial* obstruction of the airflow).

There is more to the results, however. The first of these two clusters has a noteworthy internal structure of two 'subclusters'. The first subcluster, as it were, contains all and only all bilabial phonemes whereas the second subcluster groups both alveolars together followed by a velar sound.

The second of the two big clusters also has some internal structure with two subclusters. The first of these contains all and only all nasals and liquids (i.e., phonemes that are sometimes classified as between clearcut vowels and clearcut consonants), and again the phonemes with the same place of articulation are grouped together first (the two alveolar sounds). The same is true of the second subcluster, which contains all and only all fricatives and has the labiodental fricatives merged first.

The above comments were only concerned with which elements are members of which clusters. Further attempts at interpretation may focus on how many of the clusters in Figure 70 differ from each other strongly enough to be considered clusters in their own right. Such discussion is ideally based on follow-up tests which are too complex to be discussed here, but as a quick and dirty heuristic you can look at the lengths of the vertical lines in such a tree diagram, or dendrogram. Long vertical lines indicate more autonomous subclusters. For example, the subcluster {/b/ /p/} is rather different from the remaining plosives since the vertical line leading upwards from it to the merging with {{/t/ /d/} /g/} is rather long.[50]

Unfortunately, cluster analyses do not usually yield such a perfectly interpretable output but such dendrograms are often surprisingly interesting and revealing. Cluster analyses are often used in semantic, cognitive-linguistic, psycholinguistic, and computational-linguistic studies (cf. Miller 1971, Sandra and Rice 1995, Rice 1996, and Manning and Schütze 1999: Ch. 14 for some examples) and are often an ideal means to detect patterns in large and seemingly noisy/chaotic data sets. You must realize, however, that even if cluster analyses as such allow for an objective identification of groups, the analyst must still make at least three potentially subjective decisions. The first two of these influence how exactly the dendrogram will

50. For a similar but authentic example (based on data on vowel formants), cf. Kornai (1998).

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

look like; the third you have already seen: one must decide what it is the dendrogram reflects. In what follows, I will show you how to do such an analysis with R yourself.

Hierarchical agglomerative cluster analyses typically involve the following steps:

| **Procedure** |
|---|
| Tabulating the data |
| Computing a similarity/dissimilarity matrix on the basis of a user-defined similarity/dissimilarity metric |
| Computing a cluster structure on the basis of a user-defined amalgamation rule |
| Representing the cluster structure in a dendrogram and interpreting it |

(There are many additional interesting *post hoc* tests, which we can unfortunately not discuss here.) The example we are going to discuss is from the domain of corpus/computational linguistics. In both disciplines, the degree of semantic similarity of two words is often approximated on the basis of the number and frequency of shared collocates. A very loose definition of a 'collocates of a word *w*' are the words that occur frequently in *w*'s environment, where *environment* in turn is often defined as 'in the same sentence' or within a 4- or 5-word window around *w*. For example: if you find the word *car* in a text, then very often words such as *driver*, *motor*, *gas*, and/or *accident* are relatively nearby whereas words such as *flour*, *peace treaty*, *dictatorial*, and *cactus collection* are probably not particularly frequent. In other words, the more collocates two words *x* and *y* share, the more likely there is a semantic relationship between the two (cf. Oakes 1998: Ch. 3, Manning and Schütze 2000: Section 14.1 and 15.2 as well as Gries 2009 for how to obtain collocates in the first place).

In the present example, we look at the seven English words *bronze*, *gold*, *silver*, *bar*, *cafe*, *menu*, and *restaurant*. Of course, I did not choose these words at random – I chose them because they intuitively fall into two clusters (and thus constitute a good test case). One cluster consists of three co-hyponyms of the *metal*, the other consists of three co-hyponyms of *gastronomical establishment* as well as a word from the same semantic field. Let us assume you extracted from the British National Corpus (BNC) all occurrences of these words and their content word collocates (i.e., nouns, verbs, adjectives, and adverbs). For each collocate that occurred with at least one of the seven words, you determined how often it occurred with each of the seven words. Table 44 is a schematic representation of the first

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

six rows of such a table. The first collocate, here referred to as *X*, co-occurred only with bar (three times); the second collocate, *Y*, co-occurred 11 times with *gold* and once with *restaurant*, etc.

*Table 44*. Schematic co-occurrence frequencies of seven English words in the BNC

| Collocate | *bronze* | *gold* | *silver* | *bar* | *cafe* | *menu* | *restaurant* |
|---|---|---|---|---|---|---|---|
| *X* | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| *Y* | 0 | 11 | 0 | 0 | 0 | 0 | 1 |
| *Z* | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| *A* | 0 | 0 | 0 | 1 | 0 | 2 | 0 |
| *B* | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| *C* | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| … | … | … | … | … | … | … | … |

We are now asking the question which words are more similar to each other than to others. That is, just like in the example above, you want to group elements – above, phonemes, here, words – on the basis of properties – above, average similarity judgments, here, co-occurrence frequencies. First you need a data set such as Table 44, which you can load from the file <C:/_sflwr/_inputfiles/05-5_collocates.RData>, which contains a large table of co-occurrence data – seven columns and approximately 31,000 rows.

```
> load(choose.files())·#·load·the·data·frame¶
> ls()·#·check·what·was·loaded¶
[1]·"collocates"
> attach(collocates)¶
> str(collocates)¶
'data.frame':···30936·obs.·of··7·variables:
·$·bronze····:·num··0·0·0·0·1·0·0·0·0·0·...
·$·gold······:·num··0·11·1·0·0·0·0·1·0·0·...
·$·silver····:·num··0·0·1·0·0·0·0·0·0·0·...
·$·bar·······:·num··3·0·0·1·1·1·1·0·1·0·...
·$·cafe······:·num··0·0·0·0·0·0·0·0·0·1·...
·$·menu······:·num··0·0·0·2·0·0·0·0·0·0·...
·$·restaurant:·num··0·1·0·0·0·0·0·0·0·0·...
```

Alternatively, you could load those data with `read.table(…)` from the file <C:/_sflwr/_inputfiles/05-5_collocates.txt>. If your data contain *missing data*, you should disregard those. There are no missing data, but the function is still useful to know (cf. the recommendation at the end of Chapter 2):

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> collocates<-na.omit(collocates)¶
```

Next, you must generate a similarity/dissimilarity matrix for the seven words. Here, you have to make the first possibly subjective decision, deciding on a similarity/dissimilarity measure. You need to consider two aspects: the level of measurement of the variables in point and the definition of similarity to be used. With regard to the former, we will only distinguish between binary/nominal and ratio-scaled variables. I will discuss similarity/dissimilarity measures for both kinds of variables, but will then focus on ratio-scaled variables.

In the case of nominal variables, there are four possibilities how two elements can be similar or dissimilar to each other, which are represented in Table 45. On the basis of Table 45, the similarity of two elements is typically quantified using formula (56), in which $w_1$ and $w_2$ are defined by the analyst:

$$\frac{a + w_1 \cdot d}{(a + w_1 \cdot d) + (w_2 \cdot (b + c))} \tag{56}$$

*Table 45.* Feature combinations of two binary elements

| | Element 2 exhibits characteristic *x* | Element 2 does not exhibit characteristic *x* |
|---|---|---|
| Element 1 exhibits characteristic *x* | *a* | *b* |
| Element 1 does not exhibit characteristic *x* | *c* | *d* |

Three similarity measures are worth mentioning here:

- the Jaccard coefficient: $w_1 = 0$ and $w_2 = 1$;
- the Simple Matching coefficient: $w_1 = 1$ and $w_2 = 1$;
- the Dice coefficient: $w_1 = 0$ and $w_2 = 0{,}5$.

If you define the following three vectors, what are their pairwise similarity coefficients?

```
> aa<-c(1,·1,·1,·1,·0,·0,·1,·0,·0,·0)¶
> bb<-c(1,·1,·0,·1,·0,·1,·0,·1,·0,·1)¶
> cc<-c(1,·0,·1,·1,·1,·1,·1,·1,·1,·0)¶
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

**THINK BREAK**

- Jaccard coefficient: for `aa` and `bb`: 0.375, for `aa` and `cc` 0.444, for `bb` and `cc` 0.4;
- Simple Matching coefficient: for `aa` and `bb`: 0.5, for `aa` and `cc` 0.5, for `bb` and `cc` 0.4;
- Dice coefficient: for `aa` and `bb`: 0.545, for `aa` and `cc` 0.615, for `bb` and `cc` 0.571.

But when do you use which of the three? One rule of thumb is that when the presence of a characteristic is as informative as its absence, then you should use the Simple Matching coefficient, otherwise choose the Jaccard coefficient or the Dice coefficient. The reason for that is that, as you can see in formula (56) and the measures' definitions above, that only the Simple Matching coefficient fully includes the cases where both elements exhibit or do not exhibit the characteristic in questions.

For ratio-scaled variables, there are (many) other measures, not all of which I can discuss here. I will focus on (i) a set of distance or dissimilarity measures (i.e., measures where large values represent large degrees of dissimilarity) and (ii) a set of similarity measures (i.e., measures where large values represent large degrees of similarity). The distance measures are again based on one formula and then differ in terms of parameter settings. This basic formula is the so-called Minkowski metric represented in (57).

$$(57) \qquad \left( \sum_{i-1}^{n} \left| x_{qi} - x_{ri} \right|^{y} \right)^{1/y}$$

When $y$ is set to 2, you get the so-called Euclidean distance.[51] If you insert $y = 2$ into (57) to compute the Euclidean distance of the vectors `aa` and `bb`, you obtain:

51. The Euclidean distance of two vectors of length $n$ is the direct spatial distance between two points within an $n$-dimensional space. This may sound complex, but for the simplest case of a two-dimensional coordinate system this is merely the distance you would measure with a ruler.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> sqrt(sum((aa-bb)^2))¶
[1]·2.236068
```

When *y* is set to 1, you get the so-called Manhattan- or City-Block distance of the above vectors. For `aa` and `bb`, you obtain:

```
> sum(abs(aa-bb))¶
[1]·5
```

The similarity measures are correlational measures. One of these you know already: the Pearson product-moment correlation coefficient *r*. A similar measure often used in computational linguistics is the cosine (cf. Manning and Schütze 1999: 299–303). The cosine and all other measures for ratio-scaled are available from the function `Dist` from the library `amap`.[52] This function requires that (i) the data are available in the form of a matrix or a data frame and that (ii) the elements whose similarities you want are in the rows, not in the columns as usual. If the latter is not the case, you can often just transpose a data structure (with `t`):

```
> library(amap)¶
> collocates.t<-t(collocates)¶
```

You can then apply the function `Dist` to the transposed data structure. This function takes the following arguments:

- `x`: the matrix or the data frame for which you want your measures;
- `method="euclidean"` for the Euclidean distance; `method="manhattan"` for the City-Block metric; `method="correlation"` for the product-moment correlation *r* (but see below!); `method="pearson"` for the cosine (but see below!) (there are some more measures available which I won't discuss here);
- `diag=F` (the default) or `diag=T`, depending on whether the distance matrix should contain its main diagonal or not;
- `upper=F` (the default) or `upper=T`, depending on whether the distance matrix should contain only the lower left half or both halves.

Thus, if you want to generate a distance matrix based on Euclidean distances for our collocate dataset you simply enter this:

52. The function `dist` from the standard installation of R also allows you to compute several similarity/dissimilarity measures, but fewer than `Dist` from the `library(amap)`.

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

```
> Dist(collocates.t,·method="euclidean",·diag=T,·upper=T)¶
```

As you can see, you get a (symmetric) distance matrix in which the distance of each word to itself is of course 0. This matrix now tells you which word is most similar to which other word. For example, *silver* is most similar to *cafe* because the distance of *silver* to *cafe* (2385.566) is the smallest distance that *silver* has to any word other than itself.

The following line computes a distance matrix based on the City-Block metric:

```
> Dist(collocates.t,·method="manhattan",·diag=T,·upper=T)¶
```

To get a similarity matrix with product-moment correlations or cosines, you must compute the difference 1 minus the values in the matrix. To get a similarity matrix with correlation coefficients, you therefore enter this:

```
> 1-Dist(collocates.t,·method="correlation",·diag=T,·
      upper=T)¶
···········bronze···gold·silver····bar···cafe···menu·restaurant
bronze·····0.0000·0.1342·0.1706·0.0537·0.0570·0.0462·····0.0531
gold·······0.1342·0.0000·0.3103·0.0565·0.0542·0.0458·····0.0522
silver·····0.1706·0.3103·0.0000·0.0642·0.0599·0.0511·····0.0578
bar········0.0537·0.0565·0.0642·0.0000·0.1474·0.1197·····0.2254
cafe·······0.0570·0.0542·0.0599·0.1474·0.0000·0.0811·····0.1751
menu·······0.0462·0.0458·0.0511·0.1197·0.0811·0.0000·····0.1733
restaurant·0.0531·0.0522·0.0578·0.2254·0.1751·0.1733·····0.0000
```

You can check the results by comparing this output with the one you get from `cor(collocates)¶`. For a similarity matrix with cosines, you enter:

```
> 1-Dist(collocates.t,·method="pearson",·diag=T,·upper=T)¶
```

There are also statistics programs that use 1-*r* as a distance measure. They change the similarity measure *r* (values close to zero mean low similarity) into a distance measure (values close to zero mean high similarity).

If you compare the matrix with Euclidean distances with the matrix with *r*, you might notice something that strikes you as strange …

**THINK**
**BREAK**

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

In the distance matrix, small values indicate high similarity and the smallest value in the column *bronze* is in the row for *cafe* (1734.509). In the similarity matrix, large values indicate high similarity and the largest value in the column *bronze* is in the row for *silver* (ca. 0.1706). How can that be? This difference shows that even a cluster algorithmic approach is influenced by subjective though hopefully motivated decisions. The choice for a particular metric influences the results because there are different ways in which vectors can be similar to each other.

Consider as an example the following data set, which is also represented graphically in Figure 71.

```
> y1<-1:10;·y2<-11:20;·y3<-c(6,·6,·6,·5,·5,·5,·4,·4,·4,·3)¶
> y<-t(data.frame(y1,·y2,·y3))¶
```

The question is, how similar is `y1` to `y2` and to `y3`? There are two obvious ways of considering similarity. On the one hand, `y1` and `y2` are perfectly parallel, but they are far away from each other (as much as one can say that about a diagram whose dimensions are not defined). On the other hand, `y1` and `y3` are not parallel to each other at all, but they are close to each other. The two approaches I discussed above are based on these different perspectives. The distance measures I mentioned (such as the Euclidean distance) are based on the spatial distance between vectors, which is small between `y1` and `y3` but large between `y1` and `y2`. The similarity measures I discussed (such as the cosine) are based on the similarity of the curvature of the vectors, which is small between `y1` and `y3`, but large between `y1` and `y2`. You can see this quickly from the actual numerical values:

```
> Dist(y,·method="euclidean",·diag=T,·upper=T)¶
·········y1·······y2·······y3
y1··0.00000·31.62278·12.28821
y2·31.62278··0.00000·35.93049
y3·12.28821·35.93049··0.00000
> 1-Dist(y,·method="pearson",·diag=T,·upper=T)¶
··········y1·······y2·······y3
y1·0.0000000·0.9559123·0.7796728
y2·0.9559123·0.0000000·0.9284325
y3·0.7796728·0.9284325·0.0000000
```

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

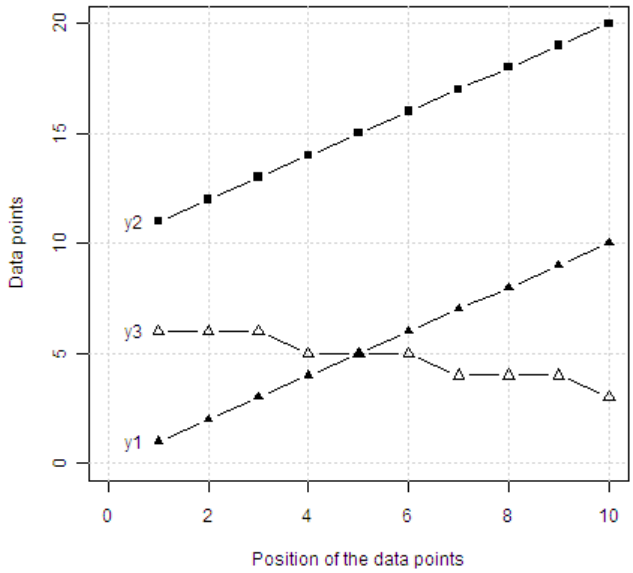


*Figure 71.* Three fictitious vectors

According to the Euclidean distance, `y1` is more similar to `y3` than to `y2` – 12.288 < 31.623 – but the reverse is true for the cosine: `y1` is more similar to `y2` – 0.956 > 0.78. The two measures are based on different concepts of similarity. The analyst must decide what is more relevant on a case-by-case basis: low spatial distances or similar curvatures. For now, we assume you want to adopt a curvature-based approach and use 1-*r* as a measure; in your own studies, you of course must state which similarity/distance measure you used, too.[53]

```
> dist.matrix<-Dist(collocates.t,·method="correlation",·diag=T,·
      upper=T)¶
> round(dist.matrix,·4)¶
···········bronze···gold·silver····bar···cafe···menu·restaurant
bronze·····0.0000·0.8658·0.8294·0.9463·0.9430·0.9538·····0.9469
gold·······0.8658·0.0000·0.6897·0.9435·0.9458·0.9542·····0.9478
silver·····0.8294·0.6897·0.0000·0.9358·0.9401·0.9489·····0.9422
bar········0.9463·0.9435·0.9358·0.0000·0.8526·0.8803·····0.7746
cafe·······0.9430·0.9458·0.9401·0.8526·0.0000·0.9189·····0.8249
menu·······0.9538·0.9542·0.9489·0.8803·0.9189·0.0000·····0.8267
restaurant·0.9469·0.9478·0.9422·0.7746·0.8249·0.8267·····0.0000
```

53. I am simplifying a lot here: the frequencies are neither normalized nor logged/dampened etc. (cf. above, Manning and Schütze 1999: Section 15.2.2, or Jurafsky and Martin 2008: Ch. 20).

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

The next step is to compute a cluster structure from this similarity matrix. You do this with the function `hclust`, which can take up to three arguments of which I will discuss two. The first is a similarity or distance matrix, and the second chooses an amalgamation rule that defines how the elements in the distance matrix get merged into clusters. This choice is the second potentially subjective decision and there are again several possibilities.

The choice `method="single"` uses the so-called *single-linkage-* or *nearest-neighbor* method. In this method, the similarity of elements *x* and *y* – where *x* and *y* may be elements such as individual consonants or subclusters such as {/b/, /p/} in Figure 70 – is defined as the minimal distance between any one element of *x* and any one element of *y*. In the present example this means that in the first amalgamation step *gold* and *silver* would be merged since their distance is the smallest in the whole matrix (1-*r* = 0.6897). Then, *bar* gets joined with *restaurant* (1-*r* = 0.7746). Then, and now comes the interesting part, {*bar restaurant*} gets joined with *cafe* because the smallest remaining distance is that which *restaurant* exhibits to *cafe*: 1-*r* = 0.8249. And so on. This amalgamation method is good at identifying outliers in data, but tends to produce long chains of clusters and is, therefore, often not particularly discriminatory.

The choice `method="complete"` uses the so-called *complete-linkage-* or *furthest-neighbor* method. Contrary to the single-linkage method, here the similarity of *x* and *y* is defined as the maximal distance between and between any one element of *x* and any one element of *y*. First, *gold* and *silver* are joined as before, then *bar* and *restaurant*. In the third step, {*bar restaurant*} gets joined with *cafe*, but the difference to the single linkage method is that the distance between the two is now 0.8526, not 0.8249, because this time the algorithm considers the maximal distances, of which the smallest is chosen for joining. This approach tends to form smaller homogeneous groups and is a good method if you suspect there are many smaller groups in your data.

Finally, the choice `method="ward"` uses a method whose logic is similar to that of ANOVAs because it joins those elements whose joining increases the error sum of squares least (which cannot be explained on the basis of the above distance matrix). For every possible amalgamation, the method computes the sums of squared differences/deviations from the mean of the potential cluster, and then the clustering with the smallest sum of squared deviations is chosen. This method is known to generate smaller clusters that are often similar in size and has proven to be quite useful in many applications. We will use it here, too, and again in your own studies, you

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

must explicitly state which amalgamation rule you used. Now you can compute the cluster structure and plot it.

```
> clust.ana<-hclust(dist.matrix,·method="ward")¶
> plot(clust.ana)¶
> rect.hclust(clust.ana,·2)·#·red·boxes·around·clusters¶
```

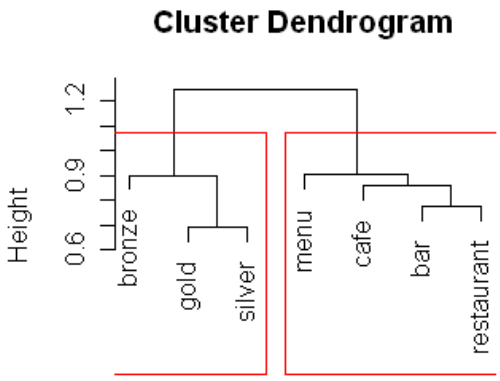


*Figure 72.* Dendrogram of seven English words

This is an uncharacteristically clearly interpretable result. As one would have hoped for, the seven words fall exactly into the two main expected clusters: one with the 'metals' and one with the gastronomy-related words. The former has a substructure in which *bronze* is somewhat less similar to the other two metals, and the latter very little substructure but groups the three co-hyponyms together before *menu* is added. With the following line you can have R show you for each element which cluster it belongs to when you assume two clusters.

```
> cutree(clust.ana,·2)¶
bronze·····gold·····silver·····bar····cafe····menu·restaurant
·····1·········1··········1·······2·······2·······2··········2
```

Now you should do the exercises for Chapter 5 …

**Recommendation(s) for further study**

- the function `daisy` (from the `library(cluster)`) to compute distance matrices for dataset with variables from different levels of measurement
- the function `kmeans` to do cluster analyses where you provide the number of clusters beforehand

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM

- the function `pvclust` (from the `library(pvclust)`) to obtain *p*-values for clusters based on resampling methods; cf. also `pvrect` and `pvpick` (from the same library)
- the function `cluster.stats` (from the `library(fpc)`), to facilitate the interpretation and validation of cluster analyses
- `Mclust` (from the `library (mclust)`) to determine the number of clusters using model-based clustering
- the function `varclus` (from the `library(Hmisc)`) to do variable clustering
- the function `nj` (from the `library(ape)`) to perform neighbor clustering and phylogenetic cluster analyses
- Crawley (2007: Ch. 23), Baayen (2008: Ch. 5), Johnson (2008: Ch. 6)

Brought to you by | ZHAW Zuercher Hochschule für Angewandte Wissenschaften
Authenticated
Download Date | 5/14/18 11:46 AM