Correspondence analysis

Exploring data and identifying patterns

Dylan Glynn
University of Paris VIII

Correspondence analysis is an exploratory technique for complex categorical data, typical of corpus-driven research. It identifies patterns of association and disassociation in those data. For instance, it can map the correlations between different uses of a linguistic form and its various social and/or morpho-syntactic contexts. The technique presents its results in the form of a two-dimensional plot, which visualises these relationships in an intuitive manner. These plots offer rich representations of the relations between different facets of complex data. Using R, this chapter explains how the technique works and offers a step-by-step explanation of its application and the interpretation of its results. The technique is also compared to the better-known and comparable cluster analysis.

Keywords: categorical data, cluster analysis, exploratory statistics, R

1. A technique for visualising correlations in categorical data

Correspondence analysis is a multivariate exploratory space reduction technique for categorical data analysis.[1] Although true, such a description tells the linguist little. Equally true, but perhaps more helpful, is to describe correspondence analysis as an exploratory technique that reveals frequency-based associations in complex corpus data. Most importantly, perhaps, the technique visualises these associations to facilitate their identification. Linguists often wish to find relations between given linguistic forms, between their meanings and in what situations those forms and meanings are used. Correspondence analysis is especially designed for identifying such usage patterning. The visualisation of the relations takes the form of configuration biplots, or maps, which depict degrees of correlation and variation through the relative proximity

1. I would like to thank Koen Plevoets who first taught me this technique and Joost van de Weijer for his help polishing this paper. All shortcomings are my own.

of data points (which represent linguistic usage features and/or the actual examples of use). This chapter describes how to perform correspondence analysis in R. It explains the R code needed to execute the analyses and shows how to interpret the results.

1.1 Use – What does correspondence analysis do?

In their quotidian research, linguists, from all kinds of theoretical orientations, analyse various usage-features of naturally occurring utterances. By way of example, imagine that one obtains 600 examples of a given word, a grammatical case, or a syntactic pattern. These examples can then be analysed, using traditional intuition-based analysis for a range of usage-features, such as tense, aspect, argument structure, agent type, the ground or path type, and the register or genre from which the example is taken. The results of analysing the examples of these usage-features can be summarised as counts of how often each of the features occurs. Significance tests can then be used to show that the occurrence of certain features is substantially more common than could be expected by chance. This statistically significant variation can then, in turn, be interpreted as representing a distinct pattern of usage.

However, with more than a couple of different dimensions of analysis or large numbers of features at play, interpreting the numbers of occurrences becomes increasingly difficult, if not impossible. Quite simply, correspondence analysis is an exploratory tool that helps one find which usage-features co-occur with other usage-features, giving a map of their overall patterning. Assuming that one is adopting a cognitive or functional approach to language, these usage-patterns can be interpreted as grammatical description, operationalised in terms of relative frequency.

It must be stressed that this technique is designed solely for exploratory purposes. In other words, it is a tool for finding things, not for establishing their significance or discerning their relevance. Therefore, it offers you no assurance that patterns found are anything more than a chance result, specific to the sample under observation. Moreover, this tool does not tell you where to look. Although exploratory, one must avoid 'fishing' for results by randomly combining factors in the hope of finding correlations that could be interpretable. Even if one finds correlations that 'make sense', such an approach increases the chance of finding co-incidental correlations or chance patterns in the sample.[2] A metaphor that might be helpful is that of the shovel for the archaeologist: if one digs randomly, everywhere, it increases the chances of finding

---

2. Weller and Romney (1990: 57) suggest performing tests for independence, such as those presented in Chapter 1, to ascertain if there exists statistically significant variation before one performs correspondence analysis. Although this is optional, it may be a good guideline to avoid falling into the trap of thinking the correlations identified are significant. It must be remembered, however, that obtaining statistical significance is dependant upon the size of the sample. With a large enough sample, very small variations will be significant, and with small

things, but exponentially increases the chance of finding irrelevant things. Correspondence analysis is a tool for digging in the data to look for patterns and correlations, but it certainly helps if one knows where to dig. This metaphor can serve us further: when an archaeologist finds an artefact, it is still up to the archaeologist to interpret the finding as well as to verify its authenticity. Correspondence analysis, assuming you have a reasonable hypothesis about where to look, is a basic and useful tool for unearthing patterns in the data, but it is no more than that.

1.2 Concept – How does correspondence analysis work?

Basically, correspondence analysis takes the frequency of co-occurring features and converts them to distances, which are then plotted, revealing how things are related by how close to or far from each other they are in a two- or three-dimensional visualisation. In the detail, there is much more to the technique, but this is the principle. Explaining a few key concepts will allow us to better understand the functioning of the technique as well as to interpret its results.

Distance matrix

The distance matrix is sometimes also called a proximity matrix and even a dissimilarity matrix. The concept is simple: the frequencies of co-occurrence are converted to distances. The resulting distance matrix can then be visualised in a two- or three-dimensional Euclidean space ('normal' perceptual space). In fact, more precisely, it is the differences between the rows and columns of frequencies that are converted to distances. Correspondence analysis uses the Chi-squared distance measure to produce the distance matrix. This measure is designed to compensate for different 'amounts' of a given category. If one has only a few examples of a given feature, let us say the 'future tense', it is highly likely that they will all, or mostly, co-occur with some other feature, such as a given verb. However, due to the low numbers involved, this is much more likely to be chance than other correlations identified. The Chi-squared distance measure attempts to compensate for this kind of bias. Nevertheless, despite the use of the Chi-squared measure, with experience, one will still observe (in the plots) correlations that are likely to be due to small numbers of a given feature. It is always necessary to go back to both the data, that is, the actual language examples, and to the raw frequencies, to see what the plots have 'revealed'. Greenacre (2007: Ch. 4) offers a lucid explanation of the Chi-squared distance measure.

---

samples (typical of semantic research), significance is much less commonly obtained. This is, of course, how it should be – it stops analysts making bold claims based on small samples.

Euclidean cloud

The distance matrix takes the form of a Euclidean cloud. In other words, it is a spread of points in a given space, like rice thrown onto a board or the holes made by darts on a dartboard. In more technical parlance, correspondence analysis computes the 'Eigenvectors' (explained below) of a correlation matrix and produces this Euclidean 'map', in one or two dimensions of that correlation. This can be thought of as reducing a set of Chi-squared scores to Euclidean distances (natural perceptual distances), suitable for two- or three-dimension visualisations. For the reader familiar with exploratory statistics, it is essentially the same as principal components analysis, but modified for categorical data.

Profiles and mass

A profile is the behavioural characteristics of a given category in the analysis, determined by the set of relative co-occurrence frequencies of that category. In a frequency (contingency) table, it is a column or a row containing all the relative frequencies of those co-occurrences. It is these profiles that the correspondence analysis plots. To calculate the profile, you add the number of occurrences for each feature in a row. You then divide each of those occurrences by the sum of them. This gives you a profile figure for each cell. The same procedure is undertaken for the columns, giving you the column profiles.

However, as mentioned above, not all the co-occurrences are of equal importance. Infrequent features would have a disproportionate effect if all were taken equally. Correspondence analysis uses the weighted averages of the profiles to compensate for this. In correspondence analysis, the term 'mass' is used to mean 'weight'. Weighting an average modifies the calculation to bias certain scores. It is widely used in basic statistics, from calculating the average score in a class test to the average monthly profit of a franchised shop. Although the idea of adding bias to any calculation may raise concern, in this case it is conventional and entirely accepted within the statistics community.

Inertia and variation

The higher the inertia one obtains, the better. Inertia is the term used in correspondence analysis to talk about the degree of variation. The inertia is calculated on observed and expected frequencies of co-occurrence. Inertia is high when column and row profiles have large deviations from their averages. In multiple correspondence analysis (as opposed to binary correspondence analysis), these scores are not normally interpretable, which is a major drawback for this form of the technique. They are not interpretable because the scores calculated seriously underestimate the amount of accurately described variation, giving unnecessarily 'bad' results. Two corrections to this have been proposed, firstly by the original author of the technique, Benzécri (1979 [reported in Greenacre 2006: 68]), and secondly by one of the current main

proponents of the technique, Greenacre (2006: 68). Greenacre argues that Benzécri's original correction was biased towards an overly optimistic result, that is, explaining more variation than was actually the case. The {ca} package, described in Section 2.3.2, includes an option to apply Greenacre's inertia adjustment.

Biplots

The concept behind the visualisation in a biplot is quite simple to understand. The correspondence analysis has calculated proximity values for the combination of the cells across the rows and columns of a contingency table. These can be plotted. Each dimension of the plot (there are two dimensions in a biplot) will represent a certain percentage of the structuring of the data variation, or 'inertia'. Plotting a single dimension, a simple line (the *x*-axis), will place the data points on this line at varying distances from each other. However, in most situations, this will poorly represent the relations between those features. If we add a second dimension, the *y*-axis, we obtain a two-dimensional biplot, typical of correspondence analysis and a range of other space reduction techniques. This will, hopefully, accurately represent a great deal of the structure in the variation in the data. Mathematically, the number of possible dimensions is equal to the number of rows or columns (whichever is smaller) minus one. So, to visualise a table with five rows and eight columns, one would need four dimensions. The scores of the inertia (or explained variation) are typically given for these first two dimensions; the *x* and *y* axes of the biplot. Although it is possible to take any two of the mathematically possible dimensions and plot these.

Normally, a combination of the first two dimensions captures a large percentage of the variation. Adding a third dimension, the *z*-axis, produces a three-dimensional plot that will even more accurately represent the behaviour of the data. Three-dimensional plots are also possible in R, but are not considered in this chapter. Sometimes, it is useful to examine combinations of dimensions one and three or even two and three in biplots, especially when the explained inertia is low. For most data sets, though, a combination of the first two dimensions offers the most accurate and interpretable visualisation of the variation and association in the data. The numerical summary of a correspondence analysis will list all the dimensions, but above the third or fourth dimension, it is rare that further dimensions represent anything more than a small fraction of the variation. In order to completely represent a contingency table, one would need all the mathematically possible dimensions.

Unfortunately, there is a range of terminology that varies from one book to another and even from one R package to the next. A few terms that may arise, especially in the numerical summaries of the analysis include: 'Eigenvalues', which indicates the inertia; the 'percentages of explained variance', or simply the percentage of inertia; and 'communalities', which are the percentages of explained inertia for individual rows or columns. If one wishes to work with the technique, there are three excellent books that explain its functioning in a clear manner, accessible even to readers with

no statistical training. These books are Greenacre (2007) *Correspondence analysis in practice*, Le Roux and Rouanet (2010) *Multiple correspondence analysis*, and Husson *et al.* (2011) *Exploratory multivariate analysis by example using R*.

1.3 Choice: Binary and multiple correspondence analysis

Correspondence analysis is, in fact, a family of techniques. There exist at least three kinds of binary correspondence analysis and three kinds of multiple correspondence analysis. Binary correspondence analysis can be understood as the basis for the multiple correspondence analysis. It has two advantages over the latter. Firstly, it indicates the percentage of explained variation for each axis. This tells you how well your analysis fits the data (how much of the variation/structure the analysis captures). It is even possible to add confidence ellipses that estimate statistical significance (see Section 2.4.1). Secondly, the contribution of each data point to the structuring of the data can be directly discerned by considering its position relative to the two axes, making the plots simpler to interpret.

On the other hand, the advantage of multiple correspondence analysis is that you can add more than two factors. The ability to capture the interaction of more than two different factors should not be underrated. In linguistics, lexical structure, syntactic structure, prosodic structure, argument structure, as well as region, gender, register and so on, are all potentially and interdependently relevant in language structure and its description. Although it is possible to combine and/or concatenate (stack) factors as one normally does for cluster analysis and binary correspondence analysis, doing so can lead to overlooking important interactions in the data. This last point is important and warrants explanation.

If we are looking at, for instance, the interaction of tense, aspect, mood and a range of near-synonymous verbs, we may propose the hypothesis that the grammatical semantics will indirectly reveal lexical semantic structure. In other words, the grammatical semantic profile of each verb will be indicative of the lexical semantic structure. To these ends, it is perfectly possible to combine the different grammatical factors, giving us usage features such as: feature 1 'present tense + indicative + perfective', feature 2: 'present tense + indicative + imperfective', feature 3: 'present tense + conditional + perfective', feature 4: 'present tense + conditional + imperfective' and so forth. However, if we then want to add further semantic or sociolinguistic features, we may miss potentially important correlations. Although combining factors in this manner will permit us to perform cluster analysis and binary correspondence analysis, we will not know if there are interactions between the different factors. We may find that, for example, the conditional mood has an important correlation with the imperfective aspect in a certain register. This may be interesting in itself, but it may also severely bias the results if not accounted for separately. For example, for this given register, is the lexical-grammatical correlation observed due to the conditional

or the imperfect or a combination? An answer to such a question is more difficult to discern in binary analysis. Therefore, there is always a trade-off – binary correspondence analysis gives more 'reliable' results and numerical indicators of explained variation, but it can struggle to represent the interaction of more than two factors simultaneously.

Other than simple binary correspondence analysis, detrended correspondence analysis and canonical correspondence analysis have been developed. Detrended correspondence analysis includes a bias added to the distance matrix calculation. It is designed to counter a well-known effect with 'long' gradients, called the 'horseshoe' effect, whereby the data points tend to form an arch. The effect is occasionally visible in plots, but any experience with correspondence analysis should avoid misinterpreting results because of it.[3] It should be noted that Greenacre (1984: 232) is sceptical about detrended correspondence analysis and it does not enjoy wide currency. It is, nevertheless, straightforward to perform in R.[4]

Canonical correspondence analysis, also termed constrained correspondence analysis, is popular in the life sciences, but is also directly relevant to linguistics. In a given study, it is perfectly common to be dealing with two different kinds of variables. Some categories interact with each other, but all relative to a different kind of category. For example, aspectual structure interacts with Aktionsart and tense in complicated and close ways. The role of, for example, register, in their interaction is of a different nature. We may not want to have the correspondence analysis treating the register features of conversation, news press and literature equally 'mixed' in with aspectual and temporal features. We can, therefore, treat the register dimension as an 'external' factor and the grammatical semantics as the 'internal' factors. The correspondence analysis then knows that we are actually interested in the internal factors and it accordingly attempts to map that space relative to the structure of the other. This results in less explained inertia overall, but (hopefully) more explained inertia for the factor that is the object of study.[5]

Multiple correspondence analysis techniques are an extension of binary correspondence analysis for the treatment of multi-way tables (binary correspondence analysis is restricted to 'normal' two-way tables). In the R sessions below, we consider three kinds – indicator matrix multiple correspondence analysis, Burt matrix multiple correspondence analysis, and joint multiple correspondence analysis.

---

3. See Greenacre (2007: 127–128) for an explanation of the horseshoe effect.

4. The package needed is {vegan} and the function is `decorana`. Oksanen (2006) has written a clear tutorial for the {vegan} package that explains the needed R command and offers examples.

5. With the package {vegan} and the function `cca`, it is straightforward to perform. Again, Oksanen (2006) offers an R tutorial on the possibilities. Canonical correspondence analysis is also performed by the {anacor} package, explained in De Leeuw and Mair (2008).

Indicator multiple correspondence analysis is sometimes called homogeneity analysis (Gifi 1990; De Leeuw and Mair 2009a). Essentially, this technique 'combines' a binary correspondence analyses using what is called a 'matrix of indicators'. Despite mathematical differences, its results are very similar to Burt matrix multiple correspondence analysis. Indeed, Greenacre (2007: 141) compares the two and concludes that there is no difference in the visualisation of the results, but does note that the Burt matrix produces more 'optimistic' percentages of inertia. However, for multiple correspondence analysis, it must be remembered that the percentages of explained inertia cannot be interpreted because they severely underestimate the representative quality of the biplot map. The Burt matrix multiple correspondence analysis is the most commonly implemented in R.

A third type of multiple correspondence analysis is based on the Burt matrix method and has been termed joint correspondence analysis. Greenacre (2006: 68; 2007: 145) argues that it is superior both in terms of the explained inertia and in the accuracy of the visualisation. It works by restricting the analysis to the cross-tabulations that typically contain the correlations of interest, those that explain the inertia. Greenacre (2007: Ch. 19) explains the technique in terms accessible to non-specialists.

2. Performing and interpreting correspondence analysis in R

Before we begin with the application *per se*, we must cover a few general questions that are relevant to every correspondence analysis. The first important question is – what to look for. There are four issues: 'fishing', over-simplicity, over-complexity, and data sparseness. Let us briefly consider each in turn.

By fishing, we mean the arbitrary (or near-arbitrary) selection of factors in the hope that one will find correlations. Correspondence analysis is a tool for identifying correlations, a tool that needs to be used in a reasoned fashion. There is no point in establishing correlations between the use of language features that bear no interpretable correlation in reality, or worse, bear an interpretable correlation, but are just a result of a few chance occurrences. In Section 1.1, the metaphor of an archaeologist digging was used to explain this point: by digging everywhere, it is sure that something will be found, but the chances of finding irrelevant things increase exponentially.

Over-simplicity is less serious a problem, but still must be borne in mind. There is no use in using correspondence analysis to identify a correlation that a simple pie chart or histogram, combined with a test for significance, would do even better. Similarly, obvious correlations can dominate results at the expense of less obvious, and therefore, more interesting results. For example, since parenthetical uses of verbs are typically in the first person, including grammatical person and parentheticality in a correspondence analysis will primarily reveal an obvious, or trivial, association. The

problem is, if these two factors are amongst a more complex range of factors, the obvious association could 'override', or 'hide', other associations. Although it is sometimes necessary to include such obvious correlations in an analysis (because one is seeking structures in other parts of the data), if it is possible to avoid doing so, then it should be avoided. Simply put, obvious correlations run the risk of 'hiding' the more interesting results. In other words, the plot will identify what is most strongly correlated instead of the subtler, yet analytically more important, correlations.

Over-complexity occurs in binary correspondence analysis when using concatenated tables (see Section 2.2.3 below) and in a multiple correspondence analysis when too many factors are examined simultaneously. For example, there is obviously no point analysing, simultaneously, 22 factors, each with 16 features, even if one has thousands of examples. Without even considering the impossibility of accounting for the variation (inertia), in such a dataset, the results would not be interpretable for the simple reason that the visualisation of so many factors becomes impossible to decipher. Moreover, the chance of 'false' associations increases dramatically with the more variables and features that are considered simultaneously. There is no steadfast rule, but thinking about how the analysis works and being realistic about its limitations are the safest ways to avoid the problem of over-complexity.

One way to avoid such over-complexity is to work with subsets. Subsets may be logical divisions within the data: for example, examining two dialects independently from one another or examining two lexemes or grammatical constructions separately. Similarly, certain features or factors can be combined. As long as the choice is reasoned and reported, it can help to simplify the interactions that the analysis is trying to explain.

This principle extends to data sparseness and 'small cells'. As a rule of thumb, one aims to have at least ten examples in each cell (the count of co-occurring features) of the cross-tabulated matrix (see below, this section). Obviously, this is not always possible, but cells of less than eight tend to cause distortions in the analysis. One may find that the analysis is 'trying so hard' to account for some relatively infrequent use that the important associations are not represented. A response to this problem is to leave out the examples (the rows in a flat data-frame, see below, this section) that contribute features only occurring a few times. First performing the correspondence analysis on the full set of data and then gradually taking out these small cells (rows of infrequent examples) is a good heuristic. Not only will it result in a better final analysis, the exploratory nature of correspondence analysis will help you to better understand the data and the correlations within them. The numerical output of binary correspondence analysis can be very helpful in identifying such problems. Using the numerical output, one can quickly see which data points are being poorly represented. This is explained in Section 2.3.1.

Let us now turn to the computation and interpretation of correspondence analysis in R. Some common packages for correspondence analysis include: {MASS}, {ca}, {languageR}, {anacor}, {homals}, {FactoMineR}, {vegan}, {ade4} and {pamctdp}. Unfortunately, for reasons of brevity, we restrict the demonstration to a small selection of functionalities in the first four of these packages. However, references to further information and tutorials on each are offered.

Each package is a suite of commands for performing correspondence analysis. They have different options and possibilities. The program R, works with functions, such as the function to read a table, to plot results of an analysis and, of course, to perform a statistical analysis. In simple terms, the functions are the commands that tell R what to do. Each function also has a set of 'arguments'. These arguments are the 'options' that R should take into account in executing the command. Moreover, keeping a record of what you have done is vital in learning to use the program. There are many additions in R for keeping your working history and also for storing the functions you use often. However, when just beginning, it is perhaps simplest to use a text file and to simply 'copy' and 'paste' to and from R. Also, at the end of an R session, it is wise to save the history (what you have done) either within R or in a separate text file. This will help you to remember the steps you took the next time you perform an analysis. Further explanation on how to use R can be found in van de Weijer and Glynn (this volume, 343–364).

In the R sessions below, after each line of command, another short line is added, following the # sign. This sign indicates that the program R should ignore what follows it and not try to interpret it as arguments belonging to the function. It is standard practice to explain command lines after such hash (#) signs.

For the purposes of explaining how to perform and interpret the analyses in R, we will use artificial data. Let us take a set of near-synonymous verbs in an imaginary language. In this language, let us say, there are three mental predicates *think*, *believe*, and *suppose*, and three communication predicates, *say*, *speak*, and *talk*, which can be used figuratively to also indicate epistemic stance, just like the mental predicates. We take 575 occurrences of the verbs, more or less equally distributed. Correspondence analysis does not require equal distribution in such a situation, but we want to have as many examples as possible of each form, so making a balanced selection is the best way of achieving this. The imaginary language possesses a three-way distinction in the aspect-mood system, distinguishing between 'Perfective', 'Imperfective' and 'Modal' forms. Each of the examples is analysed for this grammatical category. The examples are also analysed for the grammatical person of the verb and the semantic type of the indirect object. Table 1 illustrates the kind of results one expects from such an analysis.

Before we start the R session, an important aside must be made. There are two different data formats that the R functions use. It is crucial that the data is in the correct format. Details on loading the data can be found in van de Weijer and Glynn

Table 1. Example of flat data-frame

Example	Verb	Gram. category	Person	Ind. obj. semantics
example1	think	Perfective	1st	Human
example2	suppose	Modal	3rd	Concrete_Thing
example3	suppose	Perfective	3rd	Abstract_State_of_Affairs
example4	believe	Imperfective	1st	Concrete_Activity
example5	say	Imperfective	3rd	Abstract_State_of_Affairs
example6	talk	Modal	1st	Concrete_Thing
example7	suppose	Imperfective	1st	Concrete_Activity
example8	speak	Perfective	1st	Human
to 575 examples	…	…	…	…

Table 2. Example of a numerical cross-tabulation contingency table

	believe	***think***	***suppose***	***say***	***speak***	***talk***
Perfective	32	28	22	16	20	14
Imperfective	24	24	34	42	49	44
Modal	44	52	48	29	26	27

(this volume, 343–364), but this fact is essential enough that it is worth repeating. We can call one format the flat 'data-frame' and the other the numerical 'cross-tabulation' (or contingency table). The data-frame is typically what one obtains after annotating (coding) linguistic examples in Excel, Filemaker or some database application. The cross-tabulation is a result of calculating, or numerically summarising, the data-frame. This can be done in Excel with the 'pivot table' option or by using a command in R, described in van de Weijer and Glynn (this volume). The easiest way to understand the two different formats is by way of example. Table 1 is an example of a data-frame; typically it would be in a spreadsheet or in a text file with tab delimited columns. Table 2 is typical of a cross-tabulation. Note that Table 2 only cross-tabulates columns two and three: "Gram. Category" by "Verb". Again, van de Weijer and Glynn (this volume) offer more information on the different formats and how to convert data from one format to another. Once we have our data and we have the two relevant formats of the data, we can begin the correspondence analysis. Finally, note also that the data, R sessions and commands (with more detailed explanations) can be downloaded from http://dx.doi.org/10.1075/hcp.43.17gly.additional.

2.1 R package {MASS}

The package {MASS} (Venables and Ripley 2002) comes pre-installed and so it only needs to be loaded. It has simple, but effective, functions for both binary correspondence analysis and multiple correspondence analysis. The first step is to load the package:

```
> library(MASS)
```

2.1.1 *Binary correspondence analysis in MASS*

Data must first be loaded in the numerical cross-tabulated format. We take the cross-tabulation in Table 2. The first command line below loads data from a text file containing a cross-tabulation. There are three ways of loading such data into R; we will use the `choose.file()` function.

```
> data.xtab <- read.table(file.choose(),
  header= TRUE, sep="\t", row.names= 1)
  # Loads data from text file containing cross-tabulation
  # e.g.: table2.txt, downloadable from
    http://dx.doi.org/10.1075/hcp.43.17gly.additional
  # Specifies that the columns have labels, or 'headers'.
```

This command line loads the data into R, calls it 'data.xtab' and specifies that the first row in the table is the column labels, or headers (`header= TRUE`). It also assumes that any blank space, tab or otherwise, is the sign of a new field. You can use function `sep= " "` to specify how the columns are separated. This is necessary if you have labels with blank spaces in them. If you are exporting data from a spreadsheet or database, then the columns are most likely to be tab delimited, in which case add the argument `sep= "\t"`. When first beginning, it is perhaps easiest to make sure there are no blank spaces in your labels and let R guess the structure of the table. The argument `row.names= 1` specifies that the first column is the labels for the rows. This last argument is needed when loading a numerical cross-tabulation, but not for the flat data-frame.

The function for performing a binary correspondence analysis in the package {MASS} is `corresp`.

```
> ca_analysis <- corresp(data.xtab, nf= 2)
  # performs correspondence analysis on 'data.xtab'
> plot(ca_analysis)
  # plots results of 'ca_analysis'
```

Although the graphic options in R are excellent, the {MASS} package offers only a simple set of possibilities. To the last line of code above (`plot(ca_analysis)`), we

can add the graphical 'arguments' which determine the appearance of the plot.[6] For example, the argument `cex` changes the size of the font, the argument `col` specifies the colours, and `xlim` and `ylim` 'zoom' the plot by delimiting the *x* and *y* axes. These arguments should be added 'inside' the plot command, as shown in the example below:

```
> plot(ca_analysis, col= 1, cex= 1,
  xlim= c(-.225,.375), ylim= c(-.4,.4))
  # plots results of ca_analysis, in black, at font
  size 1, # 'zoomed' in on x and y-axes. See Figure 1.
```

The `col` argument can specify a range of colours by name "red", black", "blue" or by numbers, "1", "2", "3" etc. The argument `cex` takes a number that indicates the type size (1 = is default, 1.2 is larger, 1.3 larger still, and 0.8 smaller, and so forth).

Zooming can be tricky at first, but becomes simpler with practice. The plots present numbers on the *x* and *y* axes. These numbers show the distance from the centre of the plot. The *x* and *y* axes can be 'limited' both in the negative and in the positive range with the following arguments `xlim= c(-.05,.05), ylim= c(-.05,.05).` Adding this string to the plot function will make the cut-off points for the plot -0.05 and +0.05 on the *x*-axis and –0.05 and +0.05 on the *y*-axis. Change those numbers to delimit, and therefore 'zoom', the plot. Experimenting with the zoom function will allow you to get a more legible plot. In Figure 1, the data points were positioned to make maximum use of the space, that is, to make sure that the entire box was used to display the ordination of the data points.

The data submitted to this analysis are simple, but the result will allow us to understand the principles in interpreting correspondence biplots. The dispersion of the data points represents the variation of the co-occurrence of the different usage features – here, six verbs (columns) and three grammatical categories (rows). Proximity and distance represent degrees of association between the different features. The centre of the plot, indicated by the numbers on the *x* and *y* axes and by the cross in the centre, divides the plot into quadrants. This helps identify association.

The `corresp` function in {`MASS`} does not produce the most attractive plots, but since it is the simplest to perform and comes pre-installed in R, we will explain the

6. Saving plots is an important, yet often side-lined, element to using R. In Mac OSX, plots are saved in the vector format .pdf. This means one has perfect resolution magnified to infinity. Under Mac OSX, in current versions of MSWord, importing with the Insert menu (not 'cut and paste') will maintain this perfect resolution in an MSWord document. Under Windows, depending on the version, MSWord does not accept .pdf or automatically converts it with poor quality output. Under Windows, plots should be saved as .png. Although the quality is not comparable to .pdf, it is acceptable.

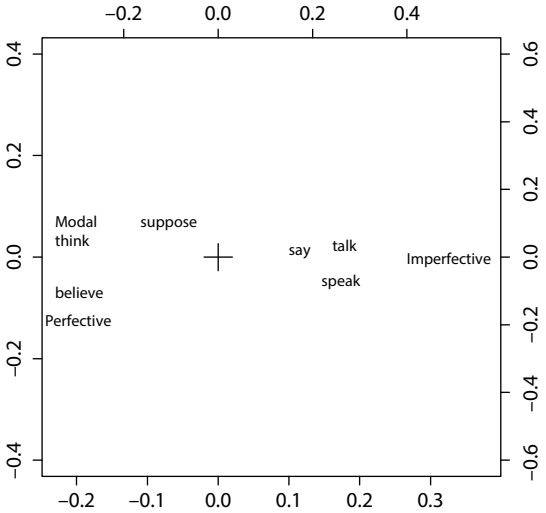

Figure 1. Binary correspondence analysis, function `corresp`, package {`MASS`}

principles of interpretation with it. In the following sections, you will see how more sophisticated, yet still simple to use, packages produce superior biplots.

In Figure 1, we see two distinct groupings of the verbs, distinguishing the mental predicates and the communication predicates. The verb *suppose* lies between the two groups, though clearly still on the same side of the plot as the other mental predicates. From this, we know that although it behaves like a mental predicate, its use is closer to the communication predicates. We also see that although the mental predicates form a distinct group, they are also distinguished along the vertical axis. The items *believe* and *think* are distinguished by the grammatical categories of Modal and Perfective. Although the distance between *think* and *believe* is relatively small, they are in different quadrants of the plot, and most importantly, the grammatical categories are on the 'far side' of the lexeme data points relative to each other. This shows the association to be distinctive. If the data points Modal and *think* were interchanged, then *believe* would be distinctly associated with Perfective, but *think* would only be associated with Modal, not distinctly so. It is for this same reason that we know there is a distinct association between the communication predicates and the Imperfective. The data point for the Imperfective lies on the 'other side' of the communication predicates data points relative to the mental predicates. This shows the distinctiveness of the Imperfective use with this group.

A note should be made about the scales printed on the axes. They are not informative on their own, but help one to gauge relative distance. This is especially important when plots are not square, but are elongated or stretched to permit the representation of all the data points. In Figure 1, we have a slightly unusual situation where the plot is skewed. In this instance, it is not interfering with the results, but if the cloud of data

points were more dense or the array more complex, this would have to be taken into account.

We can summarise the interpretation of the plot as follows:

For the mental predicates
- Distinct usage in the aspect-mood system, relative to the communication predicates
- Within the mental predicate group: *believe* is distinctly associated with the Perfective; *think* is distinctly associated with Modal; *suppose* is associated with the Modal, but its use is also relatively close to that of the communication predicates.

For the communication predicates
- Distinct in usage from the mental predicates due to their Imperfective profiling

The numerical output for the {MASS} package is not helpful at this stage. We will examine the numerical output below in Section 2.3.1.

2.1.2 *Multiple correspondence analysis in MASS*

This function is delightfully simple and powerful, but therein lies its danger. It is simple because one uses the data from a flat data-frame (such as the 'raw' data in an Excel file) which does not need to be converted to a contingency table, and also because two simple commands perform the analysis and plotting. It is powerful because (theoretically) one can add many variables to the analysis and perform complex multivariate analysis. Although this technique and these functions are excellent, care must be taken. It becomes increasingly difficult to interpret the results of the analysis as one increases the number of variables being treated. What is more, it becomes increasingly difficult to obtain reliable results.

For demonstration purposes, let us add some more fictitious data. Column 5, in Table 1, shows the semantic type of the indirect object of the utterances. We can add this factor and perform a multiple correspondence analysis. It will show the interaction of three variables, the grammatical category of aspect – mood, the lexemes, and the semantic type of the indirect object.

The function `mca` performs a multiple correspondence analysis. As mentioned above, for this function, there is no need to produce a numerical contingency table; the function accepts the flat data-frame as the input format of its data:

```
> data.frm <- read.table(file.choose(), sep="\t",
  header= TRUE)
  #loads data from text file containing data-frame
> mca_analysis <- mca(data.frm, abbrev= T)
  # performs the multiple correspondence analysis
> plot(mca_analysis, rows= F, col= 1)
  # plots results of mca_analysis, see Figure 2
```

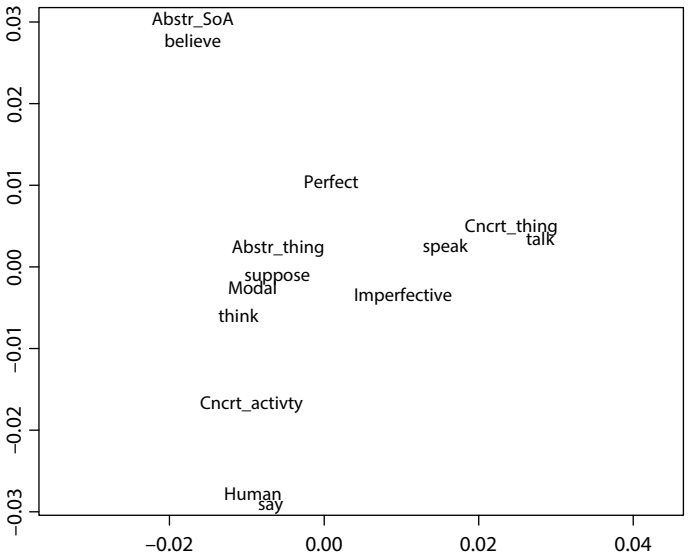

Figure 2. Multiple correspondence analysis, function `mca`, package {`MASS`}

We add `abbrev= T` (T for 'true') to the `mca` function. This tells the function not to include the factor labels. Also, note `row= F` (F for 'false'); this tells the `plot` function not to add the row numbers (that is, example numbers in linguistic analysis) to the plot. It is sometimes interesting to plot the row numbers in order to determine which actual examples are causing the interactions visualised in the plot. As for the factor labels, if one has features whose labels are the same for different factors, then one needs to add the factor labels to distinguish them on the plot. To activate these options, exchange `F` and `T` in the command line.

Figure 2 presents the results of the multiple correspondence analysis of the three factors. Firstly, you will notice that some of the data points overlap, which can cause problems for interpretation. This is a natural result of visualising association through the proximity of data points, yet it means that often one must enlarge plots (after having saved them as image files) or zoom in on them in R in order to discern what data points are overlapping. These, in turn, must be explained and described in further detail when reporting results. There exists a package, {`FactoMineR`}, which has an option for so-called dynamic graphing that allows one to displace the labels (as opposed to small data points) interactively so that they do not overlap. Details are given in Section 2.5. It is this package that was used to make the plots in descriptive studies presented in the first section of this book.

Interpreting the plot of a multiple correspondence analysis can be complex. In this plot, we have a multidimensional space that has been conflated to two dimensions. This means that data points may appear close to each other but, in fact, are

placed far apart on the oblique dimension, back 'into the page', as it were. Caution, and a little experience, makes interpreting such plots reasonably straightforward. However, as one increases the numbers of factors to four or five, the plots can only be used as a rough guide and one must return to the data to check every association identified.

In Figure 2, imagine the plot divided diagonally, horizontally and vertically. Locating the centre at the intersection of 0.00 and 0.00, we move out to see the three grammatical features, Imperfective, Perfective, and Modal, dividing up the dispersion of the plot. Once again, the Imperfective dominates the right side, where two of the communication predicates are located, *talk* and *speak*. With them is the indirect object semantic feature of Concrete Thing (Cncrt_thing). The position of the Imperfective data point, between the centre and the two communication predicates, suggests that having added the semantic features makes this less distinctly associated with the communication predicates. This interpretation would still see the Imperfective as a characteristic feature of the communication predicates, being located in the right half of the plot, but would see its distinctiveness being lessened. Although at first sight, this may seem reasonable, herein lies the trick of interpreting multiple correspondence analysis.

Another interpretation, and one more likely to be accurate, is that the Imperfective is still highly distinctive of the communication predicates, but it is being drawn to the centre by the third communication verb, *say*, which is now on the bottom left side of the plot. It is probable that there is a multiple interaction here along this dimension of use. Adding the indirect object semantics has separated *say* from the group of communication predicates. Seeing that its position on the plot almost overlaps with the indirect object semantic feature of Human, but also that these two features cluster together a long way from the centre of the plot, we can safely suppose that *say* is highly associated with a Human indirect object. We also know, from the previous analysis, that the Imperfective is highly associated with *say*. In this situation, a likely interpretation would be that both the Imperfective and the Human indirect object are associated with *say*, but that the Human Indirect object is 'pulling' the lexeme away from its Imperfective – communication verb cluster, leaving the Imperfective data point stretched between *say* and the two other lexemes, *speak* and *talk*. This happens because the Human indirect object must be highly associated with some other feature and/or highly disassociated with the other communication predicates. If this were not the case, it too would group with the Imperfective on the right of the plot. This interpretation is complex, but rich. It is also reasonably clear, if one has some experience in interpreting correspondence biplots. We will return to some of these complex associations below and reconsider these relations using different packages and different ways of handling the data.

Other findings include the distinct association of *believe* and the indirect object semantics of Abstract State of Affairs (Abstr_SoA). Also, the position of Concrete

Activity (Cncrt_activity) as object semantics, lying between *say* and *think*/*suppose*, shows that these three lexemes share this feature, yet otherwise they are distinct.

Despite this complexity, if we step back and return to our research question – the near-synonymy of the lexemes – we have a clear and coherent picture. The two sets of verbs, communication and mental, are still distinct. For the mental predicates, *believe* is distinct relative to the communication predicates. For the communication predicates, *say* is less so, relative to the mental predicates. This gives us an uneven continuum from *talk* and *speak* to *say*, which bridges the use with the mental verbs *think* and *suppose*. At the very end of this continuum, *believe* is distinctly distant from the communication predicates in its use.

Importantly, we know what characterises this continuum. At one end, the communication predicates, *speak* and *talk*, are associated with Concrete Things as indirect objects and with the Imperfective aspect. Although still associated with Imperfectivity, midway on the continuum, *say* is more associated with the indirect objects of Concrete Activity and Human. At this point, we meet the mental predicates with *suppose* and *think* sharing an association with *say* of the object type of Concrete Activity. These lexemes, *think* and *suppose*, are central to the mental predicate cluster determined by Abstract Things as indirect objects and the Perfective aspect. Finally, *believe*, also relatively close to the other mental predicates, is quite distinct due to its association with Abstract States of Affairs as object semantics.

In this fictitious description of near-synonymy, we have shown which verbs are similar and which verbs are distinct. Most importantly, we have shown why this is the case; what usage features are responsible for the similarities and differences. With some experience, such an interpretation of the results is reasonably clear. Of course, how these results inform the interpretation of language structure remains open to debate.

2.2 R package {languageR}

The {languageR} package, developed by Baayen (2011), has an impressive range of statistical options. However, for correspondence analysis, it is restricted to binary analysis. To these ends, it is simple to use and has an agreeable graphical output. The application is explained in Baayen (2008: 128–136). The package firstly needs to be installed and then loaded separately when used:

```
> library(languageR)
```

Depending on your version of R and the operating system, it may require a range of other packages to be installed and loaded before it can be loaded. The R terminal will give you instructions to follow in case this happens.

2.2.1 *Binary correspondence analysis in* `languageR`

The function for binary correspondence analysis is `corres.fnc`. It expects the data to be inputted as a numerical cross-tabulation.

```
> data.xtab <- read.table(file.choose(), header= TRUE,
  sep="\t", row.names= 1)
> data.t <- t(data.xtab)
  # transposes data.
  # Optional step, added here to improve plot legibility
> ca_analysis <- corres.fnc(data.t)
> plot(ca_analysis, ccex= 1.2, rcex= .85)
  # plots ca_analysis results with different font sizes
  # for the two axes. See Figure 3.
> title("Lexical Variation of Grammatical Categories",
  cex.main= .95, sub= "package - languageR, function -
  corres.fnc", cex.sub= 0.85)
  # Adds titles to the plot, specifies font size
```

Firstly, note that the cross-tabulation was transposed, or inverted (`data.t <- t(data.xtab)`). This is simply because some of the data point labels were 'hanging' off the edge of the plot, and transposing the table inverts the plot and thus the direction of the labels, improving legibility. The `title` function should be self-explanatory. It is entered after the `plot` line and adds the labelling to the plot. It works for most packages. There are a great many more functionalities with plotting and labelling that we do not cover here. In R, if one wishes to find what arguments (options) are available for a given function, one should type a question mark and then the function (e.g.: `?title`); a help page will appear with all the arguments available.

Interpreting Figure 3 should be straightforward, the only important difference to Figure 2 being the inversion. It is superior in its representation to the plot produced in `{MASS}`. The four quadrants are clearly indicated and the relationship between the different data points much more clearly depicted.

An important addition to this plot is the percentages indicated on each of the two axes. These percentages indicate the amount of inertia (see Section 1.2) that is explained by the first two dimensions, the plotted dimensions. It is an indication of how well the analysis is able to account for the variation in the data and is normally reported. Low inertia scores do not mean that the analysis is not valid, but it does mean that extra care should be taken in interpreting the plot. It is difficult to suggest a score that represents a 'good' level of explained inertia because it depends on the complexity of the data. Normally, in simple binary correspondence analysis, the combination of the first two dimensions should be over 75%. This will often be lower for canonical correspondence analysis and, as stressed above, for multiple correspondence analysis, the score is not interpretable.

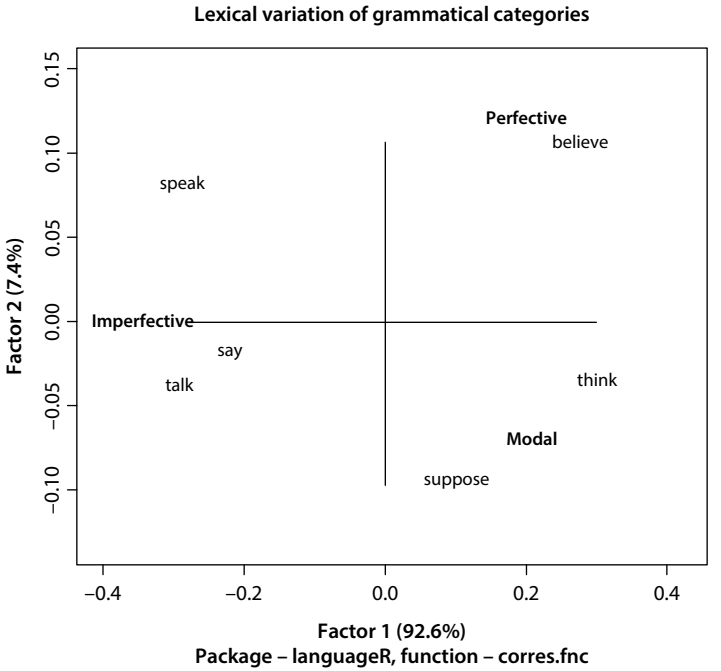

Figure 3. Binary correspondence analysis, function `corres.fnc`, package {`languageR`}

2.2.2 *Chi-squared test for significant variation*

Let us make a brief aside. In Gries (this volume, 365–390), the Chi-squared test was explained. We can apply it here to make sure that there is, indeed, significant variation between the different lexemes. The function is `chisq.test` and it expects the data to be a numerical cross-tabulation.

```
> x2.test <- chisq.test(data.xtab, correct= F)
> x2.test
data:  data
X-squared = 40.6513, df = 10, p-value = 1.301e-05
```

This shows there is significant variation between the lexemes. We can then call the Pearson residuals to see which categories are causing the most variation. Again, this is explained in Gries (this volume, 365–390).

```
> x2.test$res
           believe   think     suppose   say       speak     talk
Perfect    1.887480  0.844261 -0.383690 -0.888823 -0.387302 -1.24804
Imperfect -2.236471 -2.433997 -0.837797  1.599814  2.195817  2.104913
Modal      0.748989  1.739817  1.114179 -0.888356 -1.855656 -1.108766
```

We see from the Pearson residuals that the Imperfective uses of *speak* and *talk* and the lack of Imperfective uses of *think* and *believe* are the most important. We can bear this in mind when interpreting the plots. Looking at Figure 3, we see this fact represented visually and clearly so.

2.2.3 *Concatenating tables and combining categories*

We have, until now, worked with very simple data. Although correspondence analysis can help us find correlations in such data, its true strength is revealed when applied to more complicated and multidimensional data. In Table 1, we presented the imaginary feature analysis: there were four columns after the examples – lexeme, grammatical category, grammatical person, and indirect object semantics. We saw, in Section 2.1.2, that a multiple correspondence analysis can help examine more than two factors simultaneously, but we also saw the complexity in interpretation that arises in some data sets. Although multiple correspondence analysis is necessary for finding correlations between the factors, there are two other ways of handling data that permit some exploration of different dimensions of use.

First of all, using the pivot command in Excel, one can concatenate, or 'stack' tables. One literally generates two-dimensional tables and puts them together in a row, creating one long table. For instance, we can take the data for verb and grammatical category and, using the so-called 'pivot' function, produce a cross-tabulation. Then we repeat the operation for the verb and indirect object semantics. This gives us two tables we can join and submit to a binary correspondence analysis. However, it is much simpler to stack tables in R. A function is given in van de Weijer and Glynn (this volume, 343–364) which automatically generates stacked contingency tables in R.

Table 3 is an example of such stacked, or concatenated, results. It is perfectly acceptable to tabulate data in this manner. However, it must be remembered that when used in multivariate statistics, stacking like this can conflate conceptually different dimensions of language structure. In the instance here, the indirect object semantics and the grammatical categories of aspect and mood have been combined in such a way that any relations between these different linguistic dimensions are lost. This

Table 3. Example of concatenated cross-tabulation

Verb	Imperfect	Modal	Perfect	Abstr. SoA	Abstr. thing	Cncrt activity	Cncrt thing	Human
believe	24	44	32	41	37	8	5	9
say	42	29	16	5	25	17	8	32
speak	49	26	20	5	22	8	51	9
suppose	34	48	22	5	62	15	16	6
talk	44	27	14	1	9	4	68	3
think	24	52	28	15	37	16	13	23

Table 4. Example of concatenated cross-tabulation with combined factors

Verb + Person	Imperf.	Modal	Perfect	Abstr. SoA	Abstr. thing	Cncrt activity	Cncrt thing	Human
believe 1st Prs	18	31	12	28	23	4	3	3
believe 3rd Prs	6	13	20	13	14	4	2	6
say 1st Prs	42	15	4	3	16	10	8	24
say 3rd Prs	0	14	12	2	9	7	0	8
speak 1st Prs	44	13	9	4	16	5	36	5
speak 3rd Prs	5	13	11	1	6	3	15	4
suppose 1st Prs	28	10	0	0	22	5	9	2
suppose 3rd Prs	6	38	22	5	40	10	7	4
talk 1st Prs	37	10	8	1	7	4	42	1
talk 3rd Prs	7	17	6	0	2	0	26	2
think 1st Prs	21	11	8	3	20	3	6	8
think 3rd Prs	3	41	20	12	17	13	7	15

combination means we will only be able to see the interaction of these categories relative to the different verbs, not the interaction between them. If we are only interested in the verbs and know there are no important interactions between indirect object semantics and mood-aspect, then this poses no problems.

Table 4 shows another way of adding more information to a two-way table. Here we have combined two different factors. In Table 1, grammatical person and verb were listed as separate columns. However, semantically, it is perfectly reasonable to combine these two factors into one, a verb-person category. To obtain the new combined factor, 'sort' the two columns in Excel and add a new empty column, and create the new combined factor using the 'copy-paste' and 'change all' functions in Excel. Table 4 shows this combined factor of verb and person added to the cross-tabulation.

A point of warning should be made about data sparseness. As we add more complexity to the tables, we obtain smaller cells. The more different things we consider simultaneously, the 'thinner' the data becomes. As a rule, cells below a count of 10 should be avoided. In practice, at least when dealing with manually coded data, this rule is broken. As long as care is taken when considering the plots and it is remembered that this is only an exploratory technique, some leeway on this front can be tolerated. However, small cells should always be reported. The numerical output, which we consider below in Section 2.3.1, also helps one gauge the reliability or 'accuracy' of a data point on a biplot. Let us now consider these tables submitted to binary correspondence analysis.

Applying the same command line as that used to produce Figure 3 to the data presented in Table 3, produces the plot in Figure 4. Although the plot is inverted, the dispersion of the data points has shifted little, and the graphical representation in this package is superior; the results are the same. Indeed, the fact that the Imperfect is

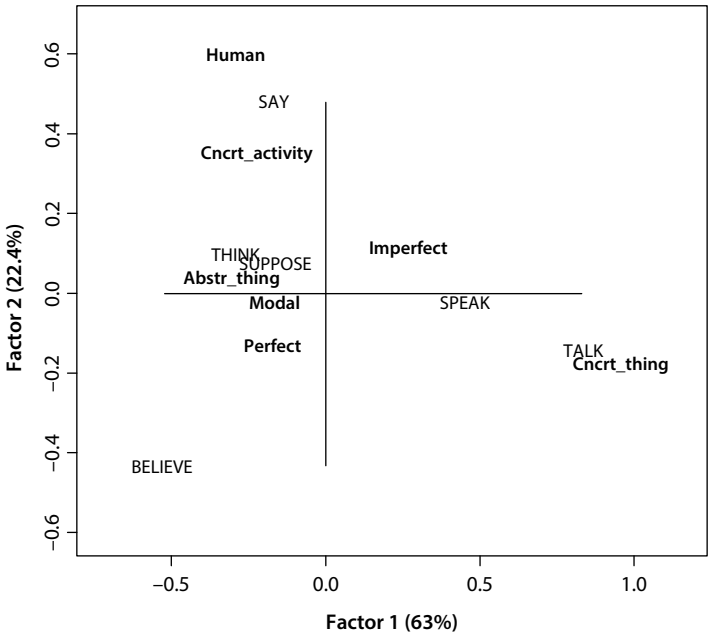

Figure 4. Binary correspondence analysis, data from Table 3, function `corres.fnc`, package {`languageR`}

being 'stretched' between two quadrants of the plot is clearer. Note also that the data point for Abstract States of Affairs lies just off the plot, distinctly and highly associated with *believe*, just as in Figure 2. It is also noteworthy that the explained inertia of the first two dimensions is 85% (dim 1: 63% + dim 2: 22.4%). This shows that the analysis is stable and we can interpret the plot with some confidence.

Figure 5 visualises yet a more complex data set, presented in Table 4. In these data, we have added the grammatical person to the verbs as well as having concatenated the grammatical categories and the indirect object semantics. The added complexity reduces the amount of explained variation, which is now just over 72% for the first two dimensions. This is still a relatively high figure and assures us that the plot remains stable, despite the added complexity.

We see that *speak* and *talk* behave in a similar manner to what we saw above in Figure 4. It seems that the addition of the variation in grammatical person does not affect their interaction with the aspect-mood and object semantics to any great extent.

However, the behaviour of *say*, which we saw was distinct in Figure 4, has been explained. The 1st person *say* is found on the Imperfective/communication verb side of the plot, but the 3rd person usage of *say*, although somewhat hidden beneath another label, is found right in the centre of a mental predicate cluster, along with the Modal and Perfective profilings. We now know that it was a simplification to understand *say* as being between the communication predicates and the mental predicates.

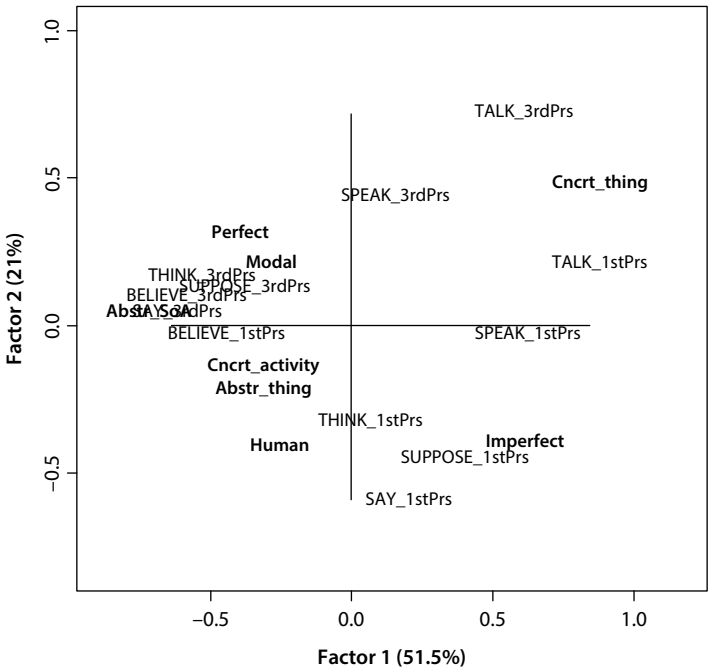

Figure 5. Binary correspondence analysis, data from Table 4, function `corres.fnc`, package {`languageR`}

It is, in fact, only the 3rd person uses that behave similarly to the mental predicates. Moreover, we see now that *believe* has joined the mental predicate cluster, which suggests that it was never so distinct from the mental predicate cluster as a whole. Instead, we see that it was just distinct from the 1st person uses of *say*, from which it was being pushed away in the visualisation. Finally, *suppose* in the 1st person has shifted right across to the Imperfective, completing the picture of a continuum between the two groups of verbs, where *suppose* in the 1st person behaves like a communication verb and *say* in the 3rd person like a mental predicate. Adding the extra dimensions of use has clarified the interaction of the verbs. It is precisely this kind of multivariate complexity that correspondence analysis is designed to capture.

If the reader wishes to perform these analyses, the command line presented in Section 2.2.1 will produce the plots.

2.3 R package {ca}

The package {ca}, developed by Nenadić and Greenacre (2007) and Greenacre and Nenadić (2010), is another commonly-used package for performing binary and multiple correspondence analysis. The package does not come with the R installation and

must be downloaded separately using the package installer. It is described in detail in Greenacre (2007: 232–240). Before we begin, load the package:

```
> library(ca)
```

This package offers a wealth of possibilities, most of which go beyond our discussion. We will focus on its numerical output and the various options for multiple correspondence analysis that it includes.

2.3.1 *Binary correspondence analysis in* `ca`

As before, the data must be loaded in a cross-tabulated matrix.

```
> data.xtab <- read.table(file.choose(), header= T,
  sep="\t", row.names= 1)
> ca_analysis <- ca(data.xtab)
> plot(ca_analysis, col= 1)
```

Due to the limitations of space, the plot of this analysis is not included. It presents the same information as above. However, the numerical output in `{ca}` is comprehensive and informative and so we will focus on this. Although most of the output does not need reporting, as one becomes more experienced with correspondence analysis, the mass and explained inertia for the individual rows and columns can help one interpret unusual patterns, especially with data sets more complex than those we are using here. There are two sets of numerical output. The first is obtained by simply typing the object of the `ca` function. We called this object `ca_analysis`. This output is not presented here because it is quite voluminous. The second numerical output is obtained by asking for a summary of the results of the analysis. The summary below is of the analysis presented in Figure 5.

```
> summary(ca_analysis)
Principal inertias (eigenvalues):
 dim    value      %   cum%   scree plot
 1      0.252400  51.5  51.5  ************************
 2      0.103030  21.0  72.5  **********
 3      0.069058  14.1  86.6  *******
 4      0.043973   9.0  95.6  ****
 5      0.016743   3.4  99.0  **
 6      0.004762   1.0 100.0
 7      00000000   0.0 100.0
        -------- -----
 Total: 0.489967 100.0
```

```
Rows:
          name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
 1 | BELIEVE_1 |  106  383  133 | -484 382  98 |  -19   1   0 |
 2 | BELIEVE_3 |   68  659   71 | -575 642  89 |   94  17   6 |
 3 |     SAY_1 |  106  737  111 |  170  56  12 | -592 681 361 |
 4 |     SAY_3 |   45  604   62 | -636 601  72 |   42   3   1 |
 5 |   SPEAK_1 |  115  953   84 |  583 953 155 |  -17   1   0 |
 6 |   SPEAK_3 |   50  760   27 |  106  42   2 |  438 718  94 |
 7 | SUPPOSE_1 |   66  651   66 |  353 254  33 | -442 397 125 |
 8 | SUPPOSE_3 |  115  524   88 | -428 485  83 |  121  39  16 |
 9 |    TALK_1 |   96  971  152 |  843 914 269 |  211  57  41 |
10 |    TALK_3 |   52  910   99 |  574 354  68 |  719 556 262 |
11 |   THINK_1 |   70  698   21 |   21   3   0 | -322 695  70 |
12 |   THINK_3 |  111  775   85 | -517 717 118 |  147  58  23 |

Columns:
          name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
 1 |      Impf |  189  958  199 |  574 638 246 | -407 320 303 |
 2 |      Modl |  197  764   73 | -311 534  76 |  204 230  80 |
 3 |      Prfc |  115  740   85 | -410 462  77 |  318 278 113 |
 4 |      A_SA |   63  427  165 | -740 425 136 |   43   1   1 |
 5 |      Abs_ |  167  408   77 | -237 248  37 | -191 160  59 |
 6 |   Cncrt_c |   59  373   33 | -279 289  18 | -151  84  13 |
 7 |   Cncrt_t |  140  998  271 |  845 753 396 |  481 244 315 |
 8 |      Humn |   71  326   97 | -223  74  14 | -411 252 117 |
```

The summary call begins with what it labels a 'scree plot'. Scree plots are used to help decide how many dimensions are needed to explain the variation in the data. In principal components analysis, factor analysis and also in multidimensional scaling, such 'plots' are common. They offer a factor-by-factor breakdown of how much variation the analysis has explained. One looks for an 'elbow' in the plot, that is, a dimension where there is a marked drop in the amount of variation explained. There are no mathematical rules to decide this point, but typically it is clear – 'most' of the variation in the dispersion of data is explained by only a few of the dimensions. The more gradual the 'descent' of the scree plot, the more trouble the analysis is having in explaining the dispersion of the data. We see here that the analysis actually lacks a clear elbow and that adding dimensions beyond the first two (the visualised dimensions) actually continues to improve the explained inertia considerably, the 3rd and 4th dimensions adding 10% explanation respectively.

The table in the summary call, 'Principal inertias', contains the vital information for understanding the structure of a correspondence analysis. It begins with the dimensions (`dim`), then lists the Eigenvalues (`value`), converts these to percentages of explained variation (`%`), and then calculates the cumulative explained variation with the addition of each dimension (`cum%`). Since biplot visualisations of the results of

a correspondence analysis typically depict the first two dimensions, the numerical output here tells us that the first two dimensions explain 72.5% of the inertia. This means the plot that we interpret does not account for just over one quarter of the variation in the data. This information is a guide to how confident we can be about the accuracy of the depiction. At 72.5%, less variation is explained than in the previous (simpler) analyses, but this figure is still sufficiently high to interpret the plot, though with some caution. The scree plot shows us that it might be informative to also visualise the third-dimension, either in a three-dimensional plot or by producing two more biplots, with dimension 1 by dimension 3 and dimension 2 by dimension 3. Many of the R packages offer these possibilities, but we do not consider them here.

Unlike `{languageR}`, the `{ca}` package does not automatically label the plots with the amount of explained inertia. If one wishes to label the *x* and *y* axes with the percentages of explained inertia, one simply uses the title function as above, for example: `title(xlab= "Dim 1 (51.5%)", ylab= "Dim 2 (21%)")`.

In the second half of the summary, we have two tables of information, one for the rows and one for the columns of the contingency table that is the basis of the analysis. Coordinates are only given for the first two dimensions (`k=1` and `k=2`), the dimensions visualised in a biplot (note the plots produced in languageR call these factors 1 and 2). Normally, in correspondence analysis, interpretation is restricted to these first two dimensions. This table breaks down the analysis for you. For each row and each column in the table, the weight assigned to that column or row is indicated (`mass`). This was explained in Section 1.2. It is essentially a bias added to the calculation to stop small numbers having a disproportionate effect.

The next score listed is the quality (`qlt`) and, as the name would suggest, this is a measure of the accuracy of the visualisation. This is a very useful score to consider. A low quality score for any given 'feature', that is, row or column, means the interpretation of its position on the plot should be treated with extra care. The numbers are given in thousandths, so a figure of 375 would be 37.5% and it indicates the explained inertia for a given row or column (that is, the labels on the plot). So, in the table above, the representation in the plot of TALK_1 would be 97.1% accurate, where BELIEVE_1 would be only 38.3% accurate. A quality score of less than 500 (50%) would suggest that the position of the data point in question does not necessarily accurately represent the relation of that feature to the others. This is often because a given feature is common to a wide range of different situations; that it correlates with distinct phenomena. For example, it may be equally used in the past tense and in the future tense, two tenses which are otherwise distinct in the analysis. In such situations, the data point will lie close to the centre of the plot, the intersection of the two axes. The other situation is mathematically similar, but analytically different. In situations where there are only few occurrences of a given feature, and those few occurrences behave in different ways, the same effect is obtained. In the latter situation, if it possible to do so without losing too much data, these examples can be omitted.

To understand why this is the case, we need to think about how the biplots work. The plot is a representation of a complex *n*-dimensional set of associations in just two dimensions. Therefore, the points of the labels are not in their original, or mathematically true, positions, having been moved to enable a two-dimensional representation. By default, the biplots present the first two dimensions, but recall that the actual number of dimensions is the number or rows or columns (whichever is less), minus one. So in the table above, we have seven dimensions (the columns, minus one). Greenacre (2007: 87) explains this in greater detail, but the principle is that the quality score here is like the inertia score, explained above, broken down for each row and column (or plot label/data point).

The inertia value (`inr`), to the right, is used to calculate the quality. But it can also be directly interpreted. The figure listed is the contribution of that row or column (feature) to explaining the total inertia. It is expressed in thousandths, so that in the output above, BELIEVE_1 explains 13.3% of the inertia in the analysis. So, given that the plot captures 72.5% of the inertia (distribution/variation in the data), this particular feature accounts for nearly 20% of the structure of the plot (13.3 / 72.5 x 100).

The next two sections of the table, to the right, give the correlation (`cor`) and the contribution (`ctr`) for each of the two dimensions. These scores are, perhaps, less useful in most circumstances, but warrant explanation. The `ctr` is the contribution that a given row or column has made to explaining the inertia along a single principal axis, that is, one of the first two dimensions. For instance, the horizontal axis (`k-1`) in Figure 5 is largely determined by three features, TALK_1, SPEAK_1, and THINK_3. The correlation scores indicate the correlation between a principal axis and the row or column in question.

2.3.2 *Multiple correspondence analysis in* `ca`

The {`ca`} package is one of the richest for multiple correspondence analysis. Its main strength lies in the fact that one can choose which of the three kinds of multiple correspondence analysis one wishes to perform. Moreover, one can choose to automatically adjust the inertias (using Greenacre's (2006) calculation for adjustment). This option is only available for the Burt matrix multiple correspondence analysis. The function for multiple correspondence analysis is `mjca`. To this, one can 'add' the argument `lambda` where one specifies the kind of multiple correspondence analysis to perform: the indicator or homogeneity analysis with `lambda= "indicator"`; the Burt matrix analysis with `lambda= "Burt"`; the joint analysis with `lambda= "JCA"`; and the Burt analysis with Greenacre adjusted inertia values with `lambda= "adjusted"`.

To perform and compare the different techniques, we use the data from the previous analyses, but we do not use the concatenated table. Multiple correspondence analysis may find important interactions between the aspect-mood grammatical semantics and the indirect object semantics, and so we must keep these two dimensions

separate. We will, however, keep the combined categories of verb and grammatical person.

For practical reasons, we cannot consider the plots of all four possibilities. The joint multiple correspondence and the adjusted inertia Burt multiple correspondence analysis produce the best results, both in terms of graphical output and explained inertia. This is to be expected since, as mentioned above, one cannot normally interpret the explained inertia scores in multiple correspondence analysis, unless they are obtained using the Greenacre 'adjusted' Burt matrix or the so-called joint method, explained below.

```
> data.frm <- read.table(file.choose(), sep="\t", header= T)
> mca_indicator_analysis <- mjca(data.frm, lambda= "indicator")
> summary (mca_indicator_analysis)

Principal inertias (eigenvalues):
 dim    value      %    cum%   scree plot
 1      0.587336  10.4   10.4  ************************
 2      0.479181   8.5   18.8  ********************
 3      0.457792   8.1   26.9  ******************
 4      0.442143   7.8   34.7  ******************
 5      0.397579   7.0   41.7  ****************
 6      0.358378   6.3   48.0  *************
 7      0.333333   5.9   53.9  ************
 8      0.333333   5.9   59.8  ************
 …
```

The output here has been abbreviated. Importantly, we see that only 18.8% of the inertia is explained. As mentioned above, inertia scores are not normally interpretable in multiple correspondence analysis and this low score is to be expected. In such analyses the inertia scores can be safely ignored. In order to obtain such scores, adjusted Burt or joint multiple correspondence analysis should be employed. These techniques are described below.

We will not consider the results of the Burt multiple correspondence analysis. It suffices to point out that the explained inertia using the Burt matrix is slightly better at 27.1%, but again this is unrealistically pessimistic. The command for the Burt multiple correspondence analysis is:

```
> mca_Burt_analysis <- mjca(data.frm, lambda= "Burt")
```

The two most promising advances in multiple correspondence analysis are certainly the joint analysis and the adjustment to inertias in the Burt matrix analysis. Let us consider these. We begin with the joint correspondence analysis (Figure 7). By not including the diagonals, which offer little information to the analysis, we greatly improve the explained inertia and also the projection onto the two-dimensional plane, the biplot. The command for the joint multiple correspondence analysis is:

```
> mca_joint_analysis <- mjca(data.frm, lambda= "JCA")
> plot(mca_joint_analysis, labels= c(0, 2),
  col= c("white", "black"))
  # plots analysis with only data point labels,
  # see Fig. 6.
```

The plot function includes two arguments that we have not seen before. The command `labels= c(0,2)` hides the row numbers (which correspond to the number of one's actual language example in the raw data set). Obviously, upon occasion, it is important to see which examples are causing the dispersion in a plot, especially when looking for exemplary occurrences in linguistic discussion and result reporting. The second argument, `col= c("white", "black")`, hides the co-occurrence points. This can be used at times to show how the co-occurrence of features is projected across the plot. For instance, sometimes there is a single kind of co-occurrence that causes a given feature to be pushed away from a group. Being able to add these points can help in plot description. Finally, the function `psch=` changes the symbol representing the point on the plot. A wide range of symbols exist and can be modified by changing the number. We cannot consider the full extent of the graphic options here, but the reader is encouraged to experiment with the plot function.

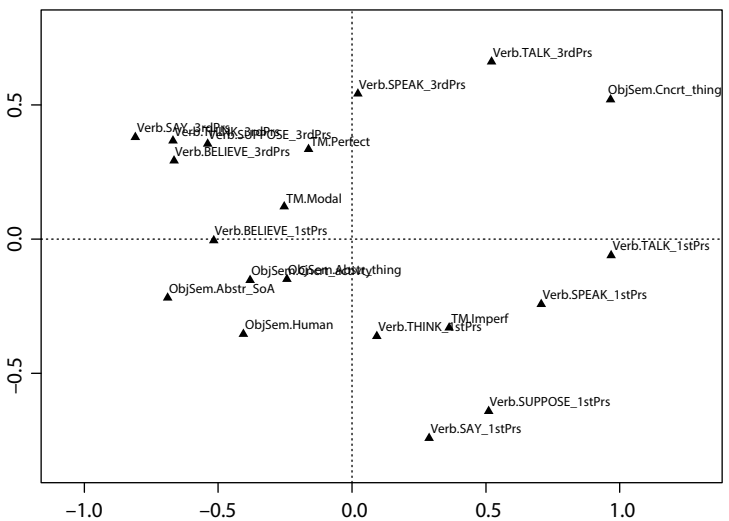

Figure 6. Joint multiple correspondence analysis, function `mjca`, package {`ca`}, method = JCA

The call for the numeric summary is as above:

```
> summary(mca_joint_analysis)
…
Diagonal inertia discounted from eigenvalues: 0.2324778
Percentage explained by JCA in 2 dimensions: 70.4%
…
```

We only consider two lines of the numerical summary of the joint correspondence analysis. As mentioned, joint analysis functions by removing the diagonals of the analysis. These 'intersections' of the tables contribute little to the explanatory power of the analysis. The first line shows us that we improved our explanation of inertia by 23% through their removal. The second line tells us that the explained inertia for the first two dimensions is 70.4%. Such a score should be reported, given the necessary caveat that estimating the explained inertia in a multiple correspondence analysis is normally unrealistically pessimistic and that this score is produced through a joint analysis.

Plot interpretation is no different to the example interpretations presented above. We will not, therefore, interpret Figure 6, but will move onto the adjusted Burt analysis. The command line follows what we saw above. We begin with a numerical summary of the analysis:

```
> mca_adjusted_analysis <- mjca(data.frm, lambda= "adjusted")
> summary(mca_adjusted_analysis)
Principal inertias (eigenvalues):
 dim    value      %   cum%   scree plot
 1      0.145165  42.0  42.0  ************************
 2      0.047861  13.8  55.8  ********
 3      0.034852  10.1  65.9  ******
 4      0.026639   7.7  73.6  *****
 5      0.009287   2.7  76.2  **
 6      0.001411   0.4  76.6
```

The above numerical summary is truncated. We see that the inertia score in the first two dimensions is 55.8%. This is relatively low, but for a multiple correspondence analysis, we can still confidently interpret the biplot. For the plotting, once again, we hide the example numbers. The command for plotting Figure 8 follows:

```
> plot(mca_adjusted_analysis, labels= c(0, 2),
  col= c("white", "black")) # Figure 7.
```

The plots in Figure 6 and Figure 7 show the same results. However, if one compares the dispersion of the data points carefully, the spread is a little clearer in Figure 7, confirming Greenacre's (2006) results in comparing the two methods. Moreover, the results here mirror those of the binary correspondence analysis presented in Figure 5.

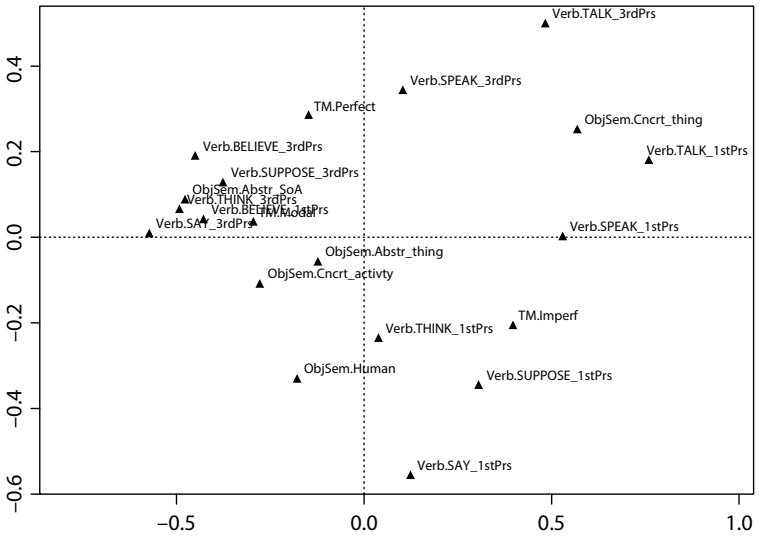

Figure 7. Burt matrix multiple correspondence analysis (with adjusted inertias), function `mjca`, package {`ca`}, method = adjusted

This shows that there are no important interactions between the factors aspect-mood and indirect object semantics. If there existed interactions between these parameters of usage, this extra complexity would be seen here.

The plot offers a coherent picture of the clustering of the mental predicates with perfective aspect and modal uses. The exceptions are the first person uses of *think* and *suppose*, which are similar to the communication predicates due to their association with the Imperfective. Relative to the mental predicates, the communication predicates *talk* and *speak* also form a cluster in the top right quadrant. This cluster is less homogenous, being based vaguely upon Concrete Things as object semantics and the Imperfect. The position of the Imperfect between the top and the bottom of the right side of the plot shows how it is drawn between the 3rd person *say* and the rest of the communication verb cluster.

2.4 R package {`anacor`}

De Leeuw and Mair (2009a) developed an excellent package for simple binary and binary canonical correspondence analysis. Although canonical analysis is a useful type of correspondence analysis, described in Section 1.3, we cannot cover the technique here. Beyond its ability to perform canonical analysis, the package offers a rich variety

of scaling and plotting options. We will consider one of these. The package must be first downloaded and at the beginning of the R session it must also be loaded:

```
> library(anacor)
```

2.4.1 *Binary correspondence analysis with confidence intervals in* `anacor`

The package {anacor} offers a range of scaling techniques and plot types – two invaluable additions to correspondence analysis. The different scaling methods can be applied to the rows and columns independently. This can be useful for revealing or highlighting different associations that are hidden in the blur of the plot. The centroid and standard scalings are typically the most useful in this regard. For some data sets, using a combination of the two dramatically increases the legibility of the results by setting one of the dimensions in the centre and 'surrounding' it with the other dimension. Unfortunately, we cannot cover this option, but the reader is strongly encouraged to experiment with combining different scaling methods with different data sets. Moreover, the package offers no fewer than seven two-dimensional plotting options. We will consider one plot type, termed the 'joint plot'. Both the plotting and scaling options are explained in detail in De Leeuw and Mair (2009a). The data are loaded as a numerical cross-tabulation:

```
> data.xtab <- read.table(file.choose(), header= T,
  sep="\t", row.names= 1)
> ca_analysis <- anacor(data.xtab, scaling=
  c("standard", "standard"))
> plot(ca_analysis, plot.type= "jointplot")
  # see Figure 8
```

The `anacor` function takes the argument `scaling=`, which specifies the scaling method for the *x* and *y*-axis. The plotting command takes the argument `plot.type=`, which specifies the type of plot desired. We have used the joint plot, which includes confidence ellipsoids. These ellipsoids are not based on a test for statistical significance, but estimate it using what the authors call the delta method (De Leeuw and Patrick 2009a). The ellipsoids are set at a 'significance' level of 95%, matching the alpha level of $p < 0.05$, standard in the social sciences.

The plot reveals the same associations as the binary plots above, but the addition of the confidence ellipsoids is a welcome advance and will prove extremely useful for some data sets. For example, it here reveals that the associations between *suppose* in the 1st person and *say* in the 1st person and the Imperfect are almost surely significant. Of course, we must return to the data for specific tests of association and/or move to configural frequency analysis and loglinear analysis for confirmatory results. The confidence ellipsoids are merely further guides to help understand relations visualised in a biplot.

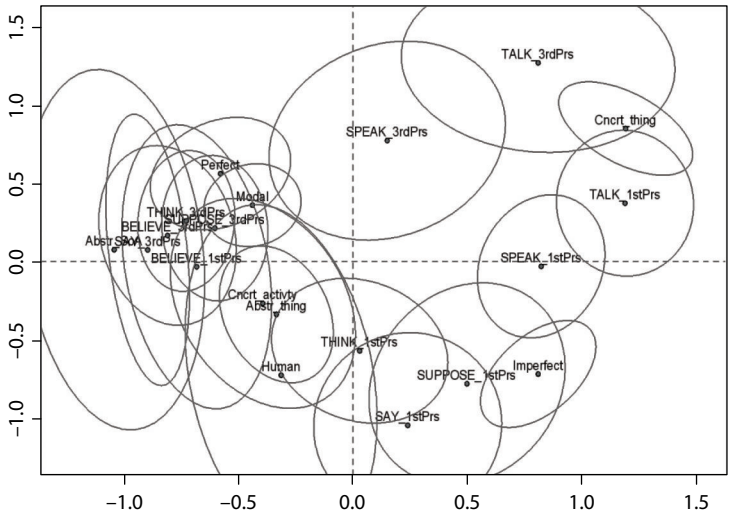

Figure 8. Binary correspondence analysis, scaling "standard" vs. "standard", function `res`, package {anacor}, plot type `jointplot`

The numerical output in `anacor` is basic but clear. The percentage of explained inertia for the first two dimensions, using the standard scaling, comes out at 82% (D1 50% + D2 32%) – a stable result:

```
> summary(ca_analysis)
z-test for singular values:
   Singular Values Asymptotical SE p-value
D1          0.5024          0.0237       0
D2          0.3210          0.0282       0
```

2.5 Other R packages for correspondence analysis

We have only treated some of the functionalities of the packages presented. However, hopefully, enough detail on both the workings of R and the application and interpretation of correspondence analysis has been covered to allow the reader to delve further into the method and the packages presented. We have omitted five packages that need to be mentioned. Armed with the explanations above, these other packages should be approachable even for people new to R.

R package {homals}

De Leeuw and Mair (2009b), the developers of the {anacor} package, presented above, also author {homals}. This package has even more powerful graphic options than {anacor}. Not only does it offer joint plots and star plots, but there is also the

option of static and interactive three-dimensional plotting. The interactive plotting allows one to turn the plot as an object in space, in order to obtain the optimal viewing point or to see how the data points are related when they are hidden behind each other. The R code is surprisingly simple, though there are large numbers of dependent packages, so make sure 'install dependencies' is selected when downloading/installing {homals}.

R package {vegan}

The {vegan} package, Oksanen (2006) and Oksanen *et al.* (2011), permits the detrended correspondence analysis described in Section 1.3. It also has excellent ordination graphics. It offers the option to 'build' correspondence plots, using a `text` function, to add labels for categories, rows, or columns, one at a time. This option is excellent for complex data sets with large numbers of categories. The analysis is performed on the entire set, but only certain data points are labelled, facilitating interpretation and reporting.

R package {ade4}

The R package {ade4} (Dray and Dufour 2007) is, in fact, an impressive suite of functions developed (and being developed) for the environmental sciences. Many of the techniques available in the package are useful for linguists. The suite includes a range of options for performing different kinds of correspondence analysis as well as for plotting not only the ordinate results (as above) but also the numerical output.

R package {FactoMineR}

A rich and powerful package, {FactoMineR} (Lê *et al.* 2008) performs principal components analysis (similar to correspondence analysis but for continuous data), binary correspondence analysis, and multiple correspondence analysis. One of its main advantages is an argument in the plot function `invisible=` which allows one to hide certain rows or columns. Although it is possible in `vegan` to build up a plot by adding rows and columns iteratively, this simple tool makes it easy to quickly see data points that are hidden, but also to remove complexity to aid in reporting. It also possesses interactive possibilities, enabling the user to move the labels on the plots (to improve legibility), to hide certain clusters of data points, and even to select certain clusters and zoom in on them. For some of these functionalities, one needs to download third party (yet free) software called DynGraph. The correspondence analyses presented in the first section of this book use this package. The use of the package is not covered in this discussion since, currently, DynGraph only runs on the MacOSX platform.

R package {pamctdp}

The package {pamctdp} (Pardo 2010) offers a range of tools for dealing with contingency tables and for controlling the rows and columns in correspondence analysis. One function that has wide application is the ability to produce barplots of the profiles

of the rows or the columns of a correspondence analysis. This simple visualisation technique can help with reporting complex data sets. Producing such plots 'manually' in R is straightforward, but time consuming.

3. Choice – correspondence or cluster

There are many statistical techniques and their number is growing. One of the most confounding hurdles for anyone beginning to use quantitative methods is knowing which techniques are possible for a given data type and which are most suitable for a given research question. This section outlines, briefly, a technique comparable to correspondence analysis and offers information on how to choose between them.

3.1 Hierarchical cluster analysis

As an exploratory technique for categorical data, correspondence analysis shares a great deal with cluster analysis (presented in the preceding chapter) in that it visualises data that are interpretable in an intuitive way by 'categorising' features relative to their occurrence with other features. Both techniques are obviously the kind of tools useful to linguists, and indeed, all social scientists.

Despite their similarities, there are important differences. The most obvious difference lies in the visual representation of the results. The graphical representation of a correspondence analysis can be more difficult to interpret than the dendrograms typical of cluster analysis, but offers important advantages over the latter.

This visual representation and interpretation is at once the strength and weakness of correspondence analysis. Firstly, the correspondence configuration biplots do not give the false impression that any observed 'clustering' is discrete, which is an unfortunate side effect of a poorly interpreted dendrogram. Of course, the discrete visualisation in a dendrogram means that the algorithm performs the 'grouping', not the interpreter (arguably more reliable). But correspondence is not about groupings; it is about associations in the data. Its representation of results is both more complex and more 'analogue', and therefore, arguably, a more socio-conceptually realistic representation of how different linguistic structures interact.

This brings us to the second, more important, advantage – the configuration biplot shows the interaction of the different features of the different factors, rather than merely sorting the features of a single factor. Correspondence analysis reveals what is associated with what, in other words, which features of which variables are co-occurring with others. This is in contrast to cluster analysis, where one only sees how the features of one factor are grouped, not what features are responsible for that grouping. In other words, the cluster analysis shows what is similar and different, but not what causes that similarity and difference.

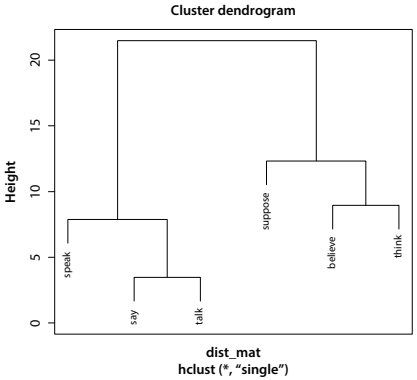

Figure 9. Hierarchical cluster analysis, dendrogram, function `hclust`, package {`MASS`}

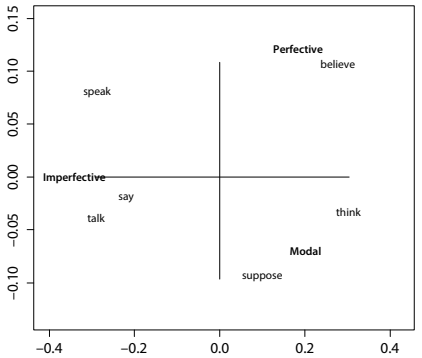

Figure 10. Binary correspondence analysis, biplot, function `corres.fnc`, package {`languageR`}

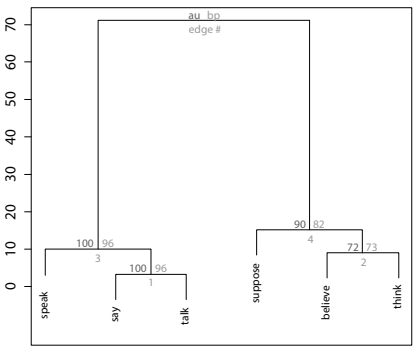

Figure 11. Hierarchical cluster analysis, dendrogram, function `pvclust`, package {`pvclust`}

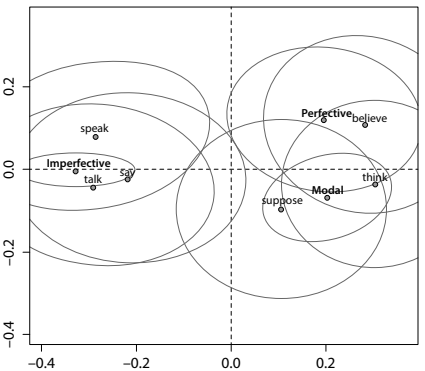

Figure 12. Binary correspondence analysis, joint plot, function `anacor`, package {`anacor`}

It should be obvious that added complexity/detail as well as an analogue/subjective interpretation can pose serious problems in correspondence analysis. The plots can be extremely difficult to read and interpret accurately. If the goal of a study is to determine which features of a given dimension are similar and which are different, cluster analysis is a simple and powerful technique. However, if the goal is to understand the interactions of different dimensions of language use and structure, then correspondence analysis is worth the extra effort.

Let us move to a comparison of the visual outputs. Figures 9–14, set out, side by side, the graphical representations resulting from the same data in the two techniques. Figure 9 is a dendrogram of a hierarchical agglomerative cluster analysis of the data

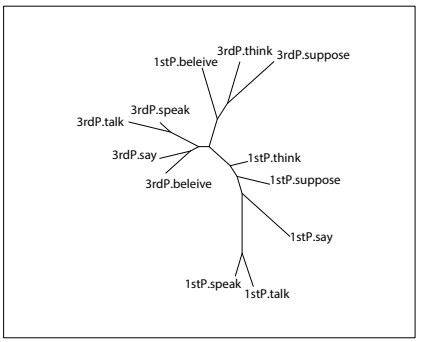

Figure 13. Hierarchical cluster analysis, phylogenetic tree, function `nj`, package {`ape`}

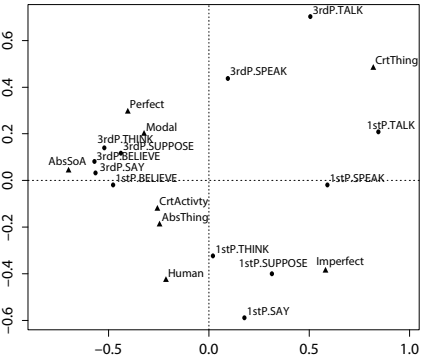

Figure 14. Multiple correspondence analysis, biplot, function `mjca`, package {`ca`}

presented in Figure 3, which is duplicated here as Figure 10 to aid in comparison. The cluster analysis in Figure 11 was performed using the package {`MASS`} and the function `hclust`. The distance matrix used is Euclidean and the agglomeration method for clustering "average". The functions expect the data to be in a numerical cross-tabulation (contingency table) format. The command line is:

```
> data.xtab <- read.table(file.choose(), header= T,
  sep= "\t", row.names= 1)
> data.dist <- dist(data.xtab, method= "euclidean")
  # converts frequency table to distance matrix
> HCA <- hclust(data.dist, method= "average")
  # performs the cluster analysis
> plot(HCA, frame.plot= T)
  # Plots results and adds frame to plot, see Figure 10.
```

The comparison between Figure 9 and Figure 10 is self-explanatory. The biplot and the dendrogram present similar information differently. The main difference is that in the biplot, we see which features cause the 'clustering'.

The dendrogram in Figure 11 is produced in the package {`pvclust`} with the function `pvclust`. It also uses the Euclidean distance matrix, but this time the agglomerating method is Ward. See Divjak and Fieller (this volume, 405–442) for an explanation of these concepts. Importantly, this function for cluster analysis includes a set of bootstrapped *p*-values to offer confidence estimates for the clusters.[7] The

7. Bootstrapping is a general statistical technique for obtaining an estimation of significance. It is not a test for significance *per se*, but is rather a method that generates a large set of samples (from the sample under investigation) with which to compare the results.

confidence scores are labelled faintly next to the branches in the clusters. The R-code for this technique follows. Again, the functions expect the data to be in a cross-tabulation format.

```
> library(pvclust)
> data.xtab <- read.table(file.choose(), header= T,
  sep= "\t", row.names= 1)
> data.t <- t(data.xtab) # inverts the data
> PVClust <- pvclust(data.t, method.hclust= "ward",
  method.dist= "euclidean")
  # produces distance matrix,
  # performs cluster analysis and bootstraps the results.
  # Can take some time to process
> plot(PVClust, frame= T)
  # Plots results, see Figure 12
```

The addition of the bootstrapped *p*-values is an important contribution to the analysis. Compare the results with Figure 12, a binary correspondence analysis with a similar estimation of significance included. The confidence ellipsoids representing significant *p*-values are designed to capture similar information. Both visualisations seem reasonably successful.

The third cluster analysis, presented in Figure 13, is not in the form of a dendrogram but of an unrooted cluster or 'phylogenetic' tree. This representation has been used by Schmidtke-Bode (2009) and Divjak (2010). This form functions poorly for only six features. Though it distinguishes them well, the sparseness of the graphics leaves a lot to be desired. Therefore, the data from Table 4 are used. The function for this kind of plot is `nj` and it is found in the {`ape`} package.

```
> library(ape)
> data.xtab <- read.table(file.choose(), header= T,
  sep= "\t", row.names= 1)
> data.dist <- dist(data.xtab)
  # converts frequency table to distance matrix
> PhyloClust <- nj(data.dist)
  # performs the cluster analysis
> plot(PhyloClust, type = "u", frame= T, cex= .9)
  # Plots results, see Figure 14
```

The phylogenetic tree in Figure 13 is compared with a multiple correspondence analysis in Figure 14. With this more complex data, we see clearly how the two methods differ in their abilities to represent data structure. The phylogenetic tree is simple and intuitive in contrast to the correspondence analysis, which needs detailed explanation.

However, we know from the discussion above that the correspondence analysis includes more information, information missing in the cluster analysis.

4. Further reading

There is a growing range of packages for performing correspondence analysis as well as new and improved tools in R generally. Moreover, new lines of statistical research are sure to bring interesting options for the technique in the near future. Within statistics, there is work on implementing permutation, or resampling, tests for correspondence analysis, as well as developing and applying mathematical algorithms that better capture certain kinds of correlations in the data – the development of grade correspondence analysis (Kowalczyk *et al.* 2004) and so-called profile-based approaches (Speelman *et al.* 2003; Delaere *et al.* submitted) are examples at hand. Within Cognitive Linguistics, broadly speaking, the method of correspondence analysis has been applied by Arppe (2006), Plevoets *et al.* (2008), Szelid and Geeraerts (2008), Glynn (2009, 2010, 2014a, 2014b, in press), Glynn and Sjölin (2011), Krawczak and Glynn (2011), Krawczak (2014a, 2014b), and Krawczak and Kokorniak (2012), as well as several studies in this volume.

There exists a good range of resources for learning more. J. P. Benzécri originally developed the technique in the late 1960s. In English, his work of reference is Benzécri (1992). More recently, Agresti (2002: 382–384) and Greenacre (1984) have championed the method. These works are difficult to approach for linguists, being concerned with the mathematics behind the technique, rather than its application and interpretation. In recent years, a range of volumes has appeared that can be used as manuals for performing the analysis. As mentioned above, Greenacre (2007) and Husson *et al.* (2011) are excellent manuals for both understanding the technique and performing it in R. Along these lines, Baayen (2008: 128–136) also offers a brief description. Le Roux and Rouanet (2010) is an excellent book: essentially it constitutes a detailed tutorial for the function-rich package {`FactoMineR`}, though it offers no R commands. Lastly, although older, Weller and Romney (1990) is another thorough and approachable alternative, but it also offers no R code. More advanced, yet still reasonably practical publications include Rencher (2002: Ch. 15), Le Roux and Rouanet (2005), and Murtagh (2005). Also consider Greenacre and Blasius (2006), which is a collection of articles that seek to advance different facets of the method. Greenacre (2007: Appendix C) offers an annotated bibliography. For specific information on the plotting options, see Gower *et al.* (2010) and Greenacre (2010).

Correspondence analysis and the current range of packages for performing it in R offer a powerful and simple tool for identifying patterns in multifactorial data. The options for visualisation of its results can be difficult to explain, but are extremely rich in the information that they display. As an exploratory method, it is an excellent heuristic for getting into complex data and digging out what relates to what.

References

Agresti, A. (2002). *Categorical data analysis* (2nd ed.). Hoboken: John Wiley. DOI: 10.1002/ 0471249688

Arppe, A. 2006. Frequency considerations in morphology. Finnish verbs differ, too. *SKY Journal of Linguistics*, 19, 175–189.

Baayen, R. H. (2008). *Analyzing linguistic data: A practical introduction to statistics using R*. Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511801686

Baayen, R. H. (2011). languageR: Data sets and functions with "Analyzing Linguistic Data: A practical introduction to statistics". R package version 1.1. Retrieved from <http://CRAN.R-project.org/package=languageR>.

Benzécri, J. P. (1992). *Correspondence analysis handbook*. New York: Marcel Dekker.

De Leeuw, J., & Mair, P. (2009a). Simple and canonical correspondence analysis using the R package anacor. *Journal of Statistical Software*, 31, 1–18. Retrieved from <http://www.jstatsoft.org/v31/i05/>.

De Leeuw, J., & Mair, P. (2009b). Gifi methods for optimal scaling in R: The package homals. *Journal of Statistical Software*, 31, 1–20. Retrieved from <http://www.jstatsoft.org/v31/i04/>.

Delaere, I., Plevoets, K., & De Sutter, G. (Submitted). Measuring text type variation through profile-based correspondence analysis: How far apart are translated and non-translated Dutch? *Target. International Journal of Translation Studies*.

Divjak, D. (2010). *Structuring the lexicon: A clustered model for near-synonymy*. Berlin & New York: Mouton de Gruyter.

Dray, S., & Dufour, A.-B. (2007). The ade4 package: Implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22, 1–20.

Gifi, A. (1990). *Nonlinear multivariate analysis*. Chichester: Wiley.

Glynn, D., & Sjölin, M. (2011). Cognitive Linguistic methods for literature: A usage-based approach to metanarrative and metalepsis. In A. Kwiatkowska (Ed.), *Texts and minds: Papers in cognitive poetics and rhetoric* (pp. 85–102). Frankfurt/Main: Peter Lang.

Glynn, D. (2009). Polysemy, syntax, and variation: A usage-based method for Cognitive Semantics. In V. Evans, & S. Pourcel (Eds.), *New directions in Cognitive Linguistics* (pp. 77–106). Amsterdam & Philadelphia: John Benjamins.

Glynn, D. (2010). Synonymy, lexical fields, and grammatical constructions: A study in usage-based Cognitive Semantics. In H.-J. Schmid, & S. Handl (Eds.), *Cognitive foundations of linguistic usage-patterns: Empirical studies* (pp. 89–118). Berlin & New York: Mouton de Gruyter.

Glynn, D. (2014a). The conceptual profile of the lexeme home: A multifactorial diachronic analysis. In J. E. Díaz-Vera (Ed.), *Metaphor and metonymy across time and cultures* (pp. 265–293). Berlin & New York: Mouton de Gruyter.

Glynn, D. (2014b). The social nature of anger: Multivariate corpus evidence for context effects upon conceptual structure. In I. Novakova, P. Blumenthal, & D. Siepmann (Eds.), *Emotions in discourse* (pp. 69–82). Frankfurt/Main: Peter Lang.

Glynn, D. (In press). Cognitive socio-semantics: The theoretical and analytical role of context in meaning. *Review of Cognitive Linguistics*.

Gower, J., Gardner-Lubbe, S., & le Roux, N. (2010). *Understanding biplots*. Chichester: Wiley.

Greenacre, M., & Blasius, J. (Eds.). (2006). *Multiple correspondence analysis and related methods*. London: Chapman & Hall. DOI: 10.1201/9781420011319

Greenacre, M., & Nenadić, O. (2010). ca: Simple, multiple and joint correspondence analysis. R package version 0.33. Retrieved from <http://CRAN.R-project.org/package=ca>.

Greenacre, M. (1984). *Theory and applications of correspondence analysis*. London: Academic Press.

Greenacre, M. (2006). From simple to multiple correspondence analysis. In M. Greenacre, & J. Blasius (Eds.), *Multiple correspondence analysis and related methods* (pp. 41–76). London: Chapman & Hall. DOI: 10.1201/9781420011319.ch2

Greenacre, M. (2007). *Correspondence analysis in practice*. London: Chapman & Hall. DOI: 10.1201/9781420011234

Greenacre, M. (2010). *Biplots in practice*. Bilbao: Fundación BBVA.

Husson, F., Lê, S., & Pagès, J. (2011). *Exploratory multivariate analysis by example using R*. London: Chapman & Hall.

Kowalczyk, T., Pleszczynska, E., & Ruland, F. (Eds.). (2004). *Grade models and methods for data analysis*. München: Springer. DOI: 10.1007/978-3-540-39928-5

Krawczak, K. (2014a). Shame and its near-synonyms in English: A multivariate corpus-driven approach to social emotions. In I. Novakova, P. Blumenthal, & D. Siepmann (Eds.), *Emotions in discourse* (pp. 84–94). Frankfurt/Main: Peter Lang.

Krawczak, K. (2014b). Epistemic stance predicates in English: A quantitative corpus-driven study of subjectivity. In D. Glynn, & M. Sjölin (Eds.), *Subjectivity and epistemicity: Corpus, discourse, and literary approaches to stance* (pp. 355–386). Lund: Lund University Press.

Krawczak, K., & Glynn, D. (2011). Context and cognition: A corpus-driven approach to parenthetical uses of mental predicates. In K. Kosecki, & J. Badio (Eds.), *Cognitive processes in language* (pp. 87–99). Frankfurt/Main: Peter Lang.

Krawczak, K., & Kokorniak, I. (2012). Subjective construal of think in Polish. *Poznań Studies in Contemporary Linguistics*, 48, 439–472. DOI: 10.1515/psicl-2012-0021

Le Roux, B., & Rouanet, H. (2005). *Geometric data analysis: From correspondence analysis to structured data analysis*. London: Kluwer.

Le Roux, B., & Rouanet, H. (2010). *Multiple correspondence analysis*. London: Sage.

Lê, S., & Husson, F. (2008). FactoMineR: An R package for multivariate analysis. *Journal of Statistical Software*, 25, 1–18.

Murtagh, F. (2005). *Correspondence analysis and data coding with R and Java*. London: Chapman & Hall. DOI: 10.1201/9781420034943

Nenadić, O., & Greenacre, M. (2007). Correspondence analysis in R, with two- and three-dimensional graphics: The ca package. *Journal of Statistical Software*, 20. Retrieved from <http://ww.jstatsoft.org/v20/i03>.

Oksanen, J., Blanchet, G., Kindt, R., Legendre, P., O'Hara, R. B., Simpson, G. L., Solymos, P., Henry, M., Stevens, H., & Wagner, H. (2011). vegan: Community ecology package. R package version 1.17-11. Retrieved from <http://CRAN.R-project.org/package=vegan>.

Oksanen, J. (2006). Multivariate analysis of ecological communities in R: vegan tutorial. Retrieved from <http://cc.oulu.fi/jarioksa/softhelp/vegan.html>.

Pardo, C. (2010). 'pamctdp'. Retrieved from <http://www.docentes.unal.edu.co/cepardot>.

Plevoets, K., Speelman, D., & Geeraerts, D. (2008). The distribution of T/V pronouns in Netherlandic and Belgian Dutch. In K. Schneider, & A. Baron (Eds.), *Variational pragmatics: Regional varieties in pluricentric languages* (pp. 181–209). Amsterdam & Philadelphia: John Benjamins.

Rencher, A. (2002). *Methods of multivariate analysis* (2nd ed.). Chichester: Wiley. DOI: 10.1002/ 0471271357

Schmidtke-Bode, K. (2009). *Going-to*-V and *gonna*-V in child language: A quantitative approach to constructional development. *Cognitive Linguistics*, 20, 509–53. DOI: 10.1515/COGL.2009.023

Speelman, D., Grondelaers, S., & Geeraerts, D. (2003). Profile-based linguistic uniformity as a generic method for comparing language varieties. *Computers and the Humanities*, 37, 317–337. DOI: 10.1023/A:1025019216574

Szelid, V., & Geeraerts, D. (2008). Usage-based dialectology: Emotion concepts in the Southern Csango dialect. *Review of Cognitive Linguistics*, 6, 23–49. DOI: 10.1075/arcl.6.03sze

Venables, W., & Ripley, B. (2002). *Modern applied statistics with S* (4th ed.). London: Springer. DOI: 10.1007/978-0-387-21706-2

Weller, S., & Romney, K. (1990). *Metric scaling correspondence analysis*. London: Sage.