Korrespondenzanalyse in R mit dem Paket „ca“

(von Georg Roth/Leipzig)

Der umfangreiche Artikel beschreibt die Funktionsweise der Korrespondenzanalyse, im Folgenden kurz CA, und ihre Anwendung mit dem Paket ‚ca’ in R anhand von simulierten Daten und Beispieldaten sowie die Ausgabe der Ergebnisgrafiken und Ergebnisdaten in externe Dateien.

1. Einführung: die CA als multivariates Verfahren

Die CA ist ein Verfahren aus dem Bereich der multivariaten Statistik. Das Ziel multivariater Statistik ist die gleichzeitige Auswertung mehrerer Merkmale, die für die Untersuchungsfälle erhoben wurden. Die für die Archäologie wichtigsten fünf Bereiche der multivariaten Statistik sind:

- das Gruppieren von Fällen anhand ähnlicher Merkmalsausprägungen mit Clusteranalysen,
- das Unterscheiden von Fällen (mit bekannter Gruppenzugehörigkeit) in Bezug auf ihre Merkmalsausprägungen mit der (linearen) Diskriminanzanalyse,
- die Beschreibung der Abhängigkeit eines (metrischen) Merkmals von anderen (metrischen) Merkmalen mittels multipler (linearer) Regression,
- die Beschreibung des Zusammenhangs jeweils eines Bündels von (metrischen) Merkmalen von einem anderen Merkmalsbündel mit der kanonischen Korrelationsanalyse,

- sowie das Anordnen von Fällen mit ähnlichen Merkmalsausprägungen in einem dimensionsreduzierten Raum – zumeist unter Bildung weniger (synthetischer) neuer Variablen (Hauptachsen) aus den bisherigen Variablen – mittels Ordination.

Zu den Ordinationsverfahren gehört neben Hauptkomponenten-, Faktoren- und Hauptkoordinatenanalysen sowie der nichtmetrischen multidimensionalen Skalierung auch die CA. Diese Methoden unterscheiden sich vor allem in Hinsicht auf die Beschaffenheit der von ihnen analysierbaren Datenstruktur.

Bei der CA basiert die Ordination zumeist auf den sog. Abundanzen. Abundanz (von ‚abundatia' latein. Reichtum/Überfluss) ist der in der Ökologie gebräuchliche Begriff für „absolute Häufigkeit". Mit der CA werden also Daten angeordnet, bei denen die Merkmalsausprägungen anhand von Auszählungen (absoluten Häufigkeiten) erfasst wurden. Aber auch Kreuztabellen binärer Merkmale (nur 0 oder 1) können mit einer CA geordnet. Die Korrespondenzanalyse, ursprünglich als ‚analyse des données' bezeichnet, geht auf den französischen Linguisten und Statistiker Jean-Paul Benzecrí (*1932) zurück, der sie Anfang der 1960er entwickelte. Im Jahr 2010 wurde die CA 50 Jahre, sowie der klassische CA-Achsenkreuzausdruck (‚biplot') und der iterative Algorithmus zur Singulärwertzerlegung – das rechnerische Verfahren hinter der modernen CA – je 40 Jahre alt.

2. Ziel und Ausgangsdaten

Das Ziel einer CA ist es, Zeilen und Spalten zweier kreuztabellierter nominaler Merkmale nach Ähnlichkeit des Auftretens von Merkmalsausprägungen anzuordnen. Das kann zum Beispiel zur Erstellung einer Relativchronologie benutzt werden. Das ist aber nur eine von zahlreichen Möglichkeiten der CA-Anwendung. Man kann eine nach ähnlicher Zusammensetzung erstellte Anordnung von Fällen ja für unterschiedlichste Phänomene erzeugen. Anwendungen reichen von der Ähnlichkeitsanordnung einzelner Quadratmeter einer Grabung nach ihrem Silexrohmaterialspektren bis zur Anordnung der archäobotanischen Unkrautartenspektren in verschiedenen Siedlungen.

In jedem Fall müssen die Zellen der als Ausgang benutzten Kreuztabelle ganze Zahlen enthalten, denn in den Tabellenzellen stehen ja Auszählungen der einzelnen Merkmalskombinationen. Auch wenn die Zeilen der Tabelle einzelne Fälle eines Untersuchungsgegenstandes, etwa die Gruben auf einem Fundplatz, repräsentieren, sind sie genau genommen ja nur Ausprägungen eines Nominalmerkmals, nämlich der Grubennummer. Ausgangspunkt ist also eine Kreuztabelle, bei der zunächst für jede Zelle der Tabelle die Abundanz des gemeinsamen Auftretens des Zeilen- und des Spaltenmerkmals ausgezählt wurde. Werden Merkmalsträger, sog. Fälle, und ein weiteres Merkmal kreuztabelliert hat die Tabelle üblicherweise folgende Form: jeder Fall ist eine Zeile dieser Kreuztabelle, jede Spalte enthält die Abundanzen einer einzigen Merkmalsausprägung. Der Name des Falls ist die Ausprägung des mit den Zeilen der Kreuztabelle dargestellten Nominalmerkmals. Die CA arbeitet also mit den Abundanzen zweier kreuztabellierter Nominalvariablen. Eine CA kann aber auch auf Kreuztabellen von binären Merkmalen angewendet werden, also Tabellen, die nur 1 und 0 enthalten und damit An- oder Abwesenheit einer Merkmalskombination erfassen.

3. Ansatz der CA

Wie misst aber nun die CA die Ähnlichkeit von Zeilen und Spalten und was ist der Bezugspunkt? Die CA verwendet eine gewichtete Variante der sog. Chiquadratdistanz.

Rechnerisch wird Ähnlichkeit, man könnte auch Nähe sagen, häufig durch ihr Gegenteil ausgedrückt, also Abstand, oder mit einem anderen Wort Distanz. Die CA misst also die Ähnlichkeit zwischen Zeilen bzw. Spalten mit ihrem Gegenteil, nämlich einer Distanz, der sog. gewichteten Chiquadratdistanz.

Ohne auf die Formel dafür genauer einzugehen kann man vereinfacht sagen, die CA misst die Unterschiede zwischen den Zeilenprozentwerten einer Zeile und den Prozenten, die in einer durchschnittlich zusammengesetzten Zeile auftreten, und gewichtet die in einer Spalte vorgefunden Unterschiede nach der Spaltensumme. Es ist also ein Maß für die Unähnlichkeit zwischen einer Zeile und der Durchschnittszeile. Dabei ist klar, dass diese Unähnlichkeit umso bedeutender ist, je größer die Summe aller Einträge in einer bestimmten Zeile ist, denn der Durchschnitt wird ja auch von den Abundanzen dieser Zeile mitgebildet. Unterscheidet sich also eine Zeile mit vielen Einträgen stark vom Durchschnitt, hat dies ein anderes Gewicht für die spätere CA-Lösung, als bei einer nur mit wenigen Einträgen besetzten Zeile. Dieser Aspekt der Bedeutung einer Zeile für die spätere CA-Lösung wird als Masse bezeichnet. Mit der gleichen Formel kann man auch die Distanz zwischen zwei Zeilen oder Spalten berechnen. Als (normalerweise nicht ausgegebenes) Zwischenergebnis der CA entsteht schließlich eine Tabelle, bei der in jeder Zelle ein Element des Chiquadratdistanzwertes steht.

Manchmal treten in Datensätzen einzelne Zeilen oder Spalten auf, die sich vom gesamten Rest der Daten stark unterscheiden. Man spricht in der Statistik dann von „Ausreißern“. Ihre Präsenz kann schwerwiegende Folgen für das CA-Ergebnis haben. Bei Abundanzen sind Ausreißer selten auftretende Merkmalskombinationen. Sie führen dazu, dass die CA zunächst vor allem zwischen ihnen und dem gesamten Rest unterscheidet. Die in allen restlichen Daten enthaltene Information wird durch solche Ausreißer verschleiert. Früher entfernten CA-Praktiker solche Datensätze einfach nach einer ersten Berechnung und wiederholten das Berechnen und Entfernen solange, bis keine stärkere Verschleierung mehr auftrat. In vielen modernen CA-Programmen ist es möglich solche Zeilen oder Spalten beizubehalten, ihnen aber bei der CA-Berechnung keine Bedeutung zuzuweisen. Sie werden dann mitgeordnet, beeinflussen das Ergebnis aber nicht mehr in unschöner Weise. Die Bezeichnung für solche Sparten ist „Ergänzungszeile/-spalte“ (‚supplementary row/column’). Indirekt wirkt sich ihre Präsenz aber selbst als Ergänzungszeile/-spalte weiterhin dadurch aus, dass jetzt die anderen Zellen einer Zeile/Spalte stärker gewichtet sind.

Die Berechnung für die Spalten wird genau auf die gleiche Weise wie bei den Zeilen erzeugt, nur dass man diesmal eben nicht entlang der Zeilen vorgeht, sondern entlang der Spalten. Die CA berechnet die Lösung für die Spalten ganz so, als ob die Tabelle entlang der Tabellendiagonale gespiegelt – man sagt transponiert – wurde. Und diese beiden Berechnungen für Zeilen und Spalten werden in einem Ergebnis vereint.

4. Berechnungsgrundlage

Die Chiquadratdistanz ist eng mit dem Chiquadrat-Test der induktiven Statistik verwandt. Würde man auf die Abundanzen-Kreuztabelle einen Chiquadrat-Test anwenden, so erhielte man den Statistikwert Chiquadrat. Teilte man diesen Wert durch die Gesamtsumme aller Abundanzen der Tabelle, so erhielte man das in der CA benützte Distanzmaß, die sog. Inertia (zu übersetzen mit ‚Trägheit’; s. u.). Die Inertia wird auch als Phi-Quadrat bezeichnet. Spielen in der CA nun also „erwartete Werte“ wie beim Testen eine Rolle und woher kommt diese Erwartungshaltung? Nein, denn man beachte, dass Erwartungswert auch ein anderer Aus-

druck für den Mittelwert einer Zufallsvariablen ist. Während im Test die Abweichung von der durchschnittlichen Zellenbelegung überprüft wird, benutzt die CA eben diese Abweichung als Messinstrument für die Unterschiedlichkeit zwischen Zeilen/Spalten untereinander und zur Durchschnittszeile/-spalte.

Das Maß für die in einer Tabelle insgesamt vorhandene Unähnlichkeit aller Zeilen (oder Spalten) zur Durchschnittszeile (oder –spalte) ist die bereits erwähnte Inertia. Man kann sich die Inertia auch als Unausgewogenheit der Tabellenzusammensetzung vorstellen: je höher die Inertia, desto mehr Unähnlichkeit liegt in der Tabelle zwischen den Zeilen (oder Spalten) und der Durchschnittszeile (oder -spalte) sowie untereinander vor – und desto erfolgreicher kann (!) eine CA die Fälle beim Anordnen auftrennen. Einzelne Zeilen (und Spalten) können unterschiedlich stark zur Gesamtinertia beitragen, was man sich bildhaft als Unwucht (Inertia) einer Zeile/Spalte vorstellen kann, die weit „außerhalb" der Durchschnittszeile/-spalte liegt. Die Inertia einer Kreuztabelle kann maximal einen Wert annehmen, der um eins kleiner ist als der kleinere der beiden Werte Zeilenanzahl oder Spaltenanzahl. Dann besteht eine Eins-zu-eins-Beziehung zwischen Zeilen und Spalten – also einzig die Zellen entlang der Diagonalen (der geordneten Tabelle - s. u. 10.) besitzen dann Einträge größer Null. Bei einer Inertia nahe Null bestehen kaum größeren Unterschiede zwischen den Zeilen (oder Spalten) untereinander. Eine hohe Inertia einzelner Zeilen/Spalten drückt sich zugleich darin aus, dass in diesen Sparten einzelne Zellen mit hohen Abundanzen weit um die am stärksten besetzte Zelle streuen.

5. Dimensionsreduktion

Was passiert da nun genau? Wenn man sich vorstellt, die (nicht ausgegebene) Zwischenergebnistabelle mit den Chiquadratdistanzwerten in ihren Zellen sei eine Koordinatentabelle, dann besäße jede Zeile Koordinaten für so viele Dimensionen, wie es Spalten gibt (und vice versa für die Spalten). Jede Zeile (oder Spalte) wäre in diesem Denkmodell ein Punkt in einem multidimensionalen Raum. Dies ist der multidimensionale Raum mit dem die CA arbeitet. Sie versucht nun wenige neue Achsen (Dimensionen) zu finden, um unter Verlust von möglichst wenig Information möglichst viel der Ausgangsinformation – die Information steckt in den multidimensionalen Koordinaten – abzubilden. Die CA sucht daher die Gerade, entlang derer die Punktwolke mit ihren „Chiquadratdistanzkoordinaten" die größte Ausdehnung besitzt. Entlang dieser Achse wird also der größte Anteil der insgesamt vorhandenen Inertia erfasst.

Ein bildliches Beispiel verdeutliche dies: Man stelle sich eine Punktwolke in der Form eines etwas platt gedrückten Zeppelins vor, der gerade steil nach Nordost startet. Ost-West, Nord-Süd und Unten-Oben sind die bisherigen drei Koordinatenachsen des Raumes, in dem wir uns den Zeppelin natürlicherweise vorstellen. Die hier Zeppelin genannte Punktwolke liegt also diagonal all zu allen drei dieser Raumachsen. Die CA sucht nun die Längsachse der Punktwolke – im Beispiel die Längsachse des Zeppelins.

Diese Längsachse vereint (meist) wesentliche Informationen von gleich mehreren bisherigen „Koordinaten-Spalten". Als nächstes wird eine weitere neue Achse durch die zeppelinförmige Punktwolke gesucht, die senkrecht zur ersten steht, und entlang derer die Punktwolke ihre zweitgrößte Ausdehnung besitzt usw. bis eine Anzahl von Geraden erreicht ist, die um eins kleiner ist als der kleinere der beiden Werte Zeilenanzahl oder Spaltenanzahl. Bei einer Tabelle mit 20 Zeilen und 10 Spalten ergeben sich also immer noch 9 Achsen. Durch die Berechnung dieser Achsen wird die ursprünglich in 10 Dimensionen (Spalten) vorliegende Information zum größten auf einigen wenigen Achsen erfasst: dies nennt man Dimensionsreduktion.

6. CA-Achsen

Die neu erzeugten CA-Geraden oder CA-Achsen kreuzen sich in einem Punkt, dem Nullpunkt des späteren Ordinationsergebnisses. Die Ausdrücke Gerade, Achse, Hauptachse oder Dimensionen werden hier als Synonyme verwendet. Da mit der gewichteten Chiquadratdistanz der Abstand zur durchschnittlichen Zeile oder Spalte gemessen wird, ist leicht einsichtig, dass eine völlig dem Durchschnitt entsprechende Zeile oder Spalte genau im Nullpunkt des Achsenkreuzes liegt. Auf diesen neuen Achsen lässt sich für jeden Punkt eine Koordinate angeben, indem man das Lot vom Punkt auf die entsprechende Achse fällt. Da die erste Achse der Längsachse der Punktwolke entspricht, auf ihr also wesentlich mehr Unterschiede zwischen den Punkten erfasst sind, ist sie (in der Regel) wichtiger als die zweite Achse. Analoges gilt für die weiteren Achsen. Bei der Berechnung der Achsenausrichtungen haben stärker besetzte Zeilen/Spalten einen stärkeren Einfluss auf deren Orientierung, da größere Abundanzen ja mehr Bedeutung für das Zustandekommen der Durchschnittszeile/spalte besitzen. Die Besetzung (Masse) der Zeilen/Spalten wirkt also wie eine Art Anziehungskraft auf die Achsen. Im obigen ‚Zeppelinbeispiel' kann man sich die Masse einfach als unterschiedliche Größe der Punkte vorstellen. Zur Wiederholung, ein Punkt, der im Nullpunkt (Ursprung) der CA-Achsen liegt, entspricht in etwa der Durchschnittszeile bzw. -spalte.

Ist die CA-Lösung jetzt die Vereinfachung der Datenstruktur, die man gesucht hat? Nun immerhin erfassen die ersten paar Achsen (in der Regel) deutlich mehr Unterschiede als die folgenden und mit diesem Umstand begründet sich, dass man anstatt aller CA-Achsen meist nur die ersten zwei oder drei betrachtet, da sie ja wesentlich mehr Information über die (Un-)Ähnlichkeit enthalten als die anderen. Die Bedeutung der Achsen misst man in der Regel anhand des Anteils der Gesamtinertia, der von ihnen erfasst wird – man sagt auch, der auf ihnen abgebildet ist. Natürlich kann man auch den Absolutbetrag der von ihnen erfassten Inertia angeben.

Rechnerisch wird das Vorgehen der „Achsensuche" bei den meisten Programmen (und alle R-Paketen) mittels der sog. Singulärwertzerlegung (engl. ‚singular value decomposition', kurz ‚svd') der Tabelle durchgeführt, die in ihren Zellen die "Chiquadratdistanzkoordinaten" enthält. Diese Tabelle wird wie gesagt bei der Berechnung der CA normalerweise nicht ausgegeben. Dabei handelt es sich um eine Matrixalgebra-Berechnung. Die früher häufiger verwandte rechnerische Lösung mit dem Wiederholen der folgenden Arbeitsschritte, Durchschnittszeilen/-spalten-Bildung, Berechnen der gewichteten Chiquadratdistanzen und Neuanordnung von Zeilen- und Spalten, das sog. ‚reciprocal averaging', findet in keiner moderneren CA-Software mehr Anwendung. Die SVD ist rechnerisch genauer. Sie findet die Koordinaten der CA-Lösung durch die Lösung linearer Gleichungssysteme.

7. CA-Berechnung

An dieser Stelle noch ein kleiner Exkurs zum besseren Verständnis: Das Drehen (und Strecken bzw. Stauchen) von Punktwolken wird rechnerisch durch die Multiplikation der Koordinatentabelle mit einer weiteren, quadratischen Tabelle umgesetzt, die so viel Zeilen und Spalten hat, wie die Koordinatentabelle Spalten. Bildhaft kann man sich das – etwas unzulässig vereinfacht – im Fall der SVD so vorstellen: würde man mit der Chiquadratdistanztabelle eine Punktwolke drehen, so wären die CA-Achsen die Rotationsachsen der Punktwolke, die CA-Koordinaten das Drehergebnis und die Singulärwerte die Streckungs- bzw. Stauchungsfakto-

ren. Aufgrund der Eigenschaften der SVD wäre die erste Rotationsachse zugleich die Längsachse der Ausgangspunktwolke. Hier ist ein wichtiger Aspekt vorzumerken: Beachtet man bei der Grafikausgabe der Punkte auf den CA-Achsen also die Singulärwerte nicht, so erhält man nur eine Standardlösung und es erfolgt keine Streckung oder Stauchung. Diese Standardlösung ordnet die Zeilen nach ihrer Ähnlichkeit zur Durchschnittszeile (und untereinander) an, und die Spalten nach ihrer Ähnlichkeit zur Durchschnittsspalte (und untereinander).

Die pro Achse erfasste Inertia (Unähnlichkeit) wird mit dem jeweiligen Singulärwert der Achse gemessen – also dem Streckungs-/Stauchungsfaktor der rotierten Punktwolke. Erwartungsgemäß ist also der Singulärwert der ersten Achse der größte. Der Singulärwert der zweiten Achse ist der zweitgrößte und so fort. Die Summe der Singulärwerte ist ein Maß der insgesamt vorhandenen Inertia. Teilt man den Singulärwert einer Achse durch die Summe aller Singulärwerte, erhält man den Anteil der von der Achse erfassten Streuung – quasi den Prozentsatz der auf dieser Achse erfassten Unähnlichkeit (Inertia).

8. Koordinatenskalierung

Bei der späteren grafischen Ergebnisausgabe ist zu bedenken, dass die nach der Standardlösung erzeugten Zeilen- und Spaltenkoordinaten nicht in einem gemeinsamen Raum liegen! Schließlich entspricht bei der Darstellung der Zeilen der Nullpunkt des Achsenkreuzes der Position der Durchschnittszeile, bei der Darstellung der Spalten aber ist der Nullpunkt die Position der Durchschnittsspalte – das sind natürlich unterschiedliche Dinge. Dieses Dilemma wird so gelöst, dass man (normalerweise) die Achsen-Koordinatenwerte der Zeilen mit den Singulärwerten multipliziert. Da diese aus rechnerischen Gründen immer kleiner als 1 sind, werden die Zeilen in den Raum der Spaltenstandardlösung ‚hineingestaucht'.

Die Abbildung dieser Zeilenkoordinaten und der Spaltenstandardlösungskoordinaten ergibt eine Art Raum, bei der jeweils ein Spaltenpunkt dort liegt, wo ein Zeilenpunkt läge, der nur Ausprägungen dieser Spalte aufweist. Das Umrechnen der CA-Koordinaten wird als Skalierung bezeichnet. Die durch die Multiplikation mit den Singulärwerten gestauchten Zeilenkoordinaten werden auch als ‚Zeilenprinzipalkoordinaten' bezeichnet. Nicht umgerechnete Koordinaten bezeichnet man als Standardkoordinaten.

Eine Darstellung mit jeweils einer Kategorie in Standard- und einer in Prinzipalkoordinaten hat mehrere interpretatorische Vorteile. Bei der als ‚symmetric biplot' bezeichneten grafischen Abbildung der ersten beiden Hauptachsen bedeutet die Nachbarschaft zweier Zeilenpunkte eine ähnliche Zusammensetzung der Zeileninventare, die Nachbarschaft zweier Spaltenpunkte bedeutet, dass die Spaltenmerkmale häufig in den gleichen Zeilen zusammen auftreten. Wenn man nun die Darstellungsart mit den Zeilen (oder Spalten) in Prinzipalkoordinaten gewählt hat, darf man auch die Nähe von Zeilenpunkten zu den Spaltenpunkten interpretieren: je näher ein Zeilenpunkt einem Spaltenpunkt auf der gleichen Seite des Nullpunktes ist, desto überdurchschnittlich häufiger ist die in der entsprechenden Spalte erfasste Merkmalsausprägung in dieser Zeile vertreten. Liegt der Spaltenpunkt dagegen von einem Zeilenpunkt aus gesehen auf der anderen Seite des Nullpunktes, ist die entsprechende Merkmalsausprägung unterdurchschnittlich in dieser Zeile vertreten. Die randlich in der Ordinationslösung angeordneten Zeilenpunkte, werden in ihrer Zusammensetzung von einzelnen benachbarten Spaltenmerkmalen auffällig dominiert.

Bei vielen herkömmlichen Programmen ist allerdings nicht klar, welche Koordinaten der CA-Achsenkreuzausdruck überhaupt abbildet. Eine Stärke der meisten R-Pakete zur CA ist eine genaue Steuerung der Skalierung des CA-Ergebnisausdruckes. Neben der hier empfohlenen Zeilenprinzipalkoordinatenvariante bieten einige Pakete auch noch andere Skalierungen an, deren rechnerische Qualitäten zwar interessant, deren Deutung allerdings wesentlich unverständlicher ist.

9. CA zur Chronologieerzeugung und der „Parabeltest“

Die Grundannahme für die Benutzung der CA als Verfahren zur Erzeugung einer relativen Chronologie betrifft die Häufigkeit der Typen, anhand derer man die Chronologie erzeugen will. Man nimmt an, dass ein Typ mit der Zeit zunächst immer häufiger auftritt und nach Erreichen eines Maximums dann wieder seltener wird, bis er aus der Mode kommt. Im Fall von drei Typen, einem alten, einem mittleren und einem jungen wäre folgende Verteilung über Befunde zu erwarten. In alten Befunden sollte der älteste Typ häufig, der mittlere selten und der jüngste abwesend sein. In mittelalten Befunden sollte der ältere und der jüngere Typ selten, der mittlere aber häufig sein und in jüngeren Befunden sollte der ältere Typ abwesend, der mittlere selten und der jüngste häufig sein.

Typ/Spalten - Befund/Zeilen

alt	mittel	jung	
3	1	0	alt
1	3	1	mittel
0	1	3	jung

Wenn diese Annahme zutrifft, ergibt eine mittels CA diagonalisierte Abundanzentabelle die zeitliche Reihenfolge von Typen und Befunden (s. u. 10.). Würde aber ein Typ mit der Zeit häufiger, dann seltener und schließlich wieder häufiger, dann versagt die CA bei der Relativchronologie dieses Typs.

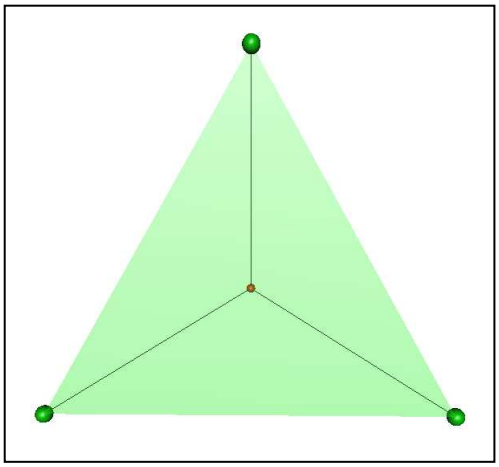
Folgt die Häufigkeit der Merkmalsauspägungen, ihre regelmäßige Zu- und Abnahme, der Veränderung externer Kausalphänomene spricht man von Gradienten. Im Fall der Relativchronologieanwendung nimmt man an, dass das wichtigste Kausalphänomen, der sog. Gradient, also die Zeit ist. Wenn das CA-Ergebnis sich mit bekannten Chronologieansätzen vereinbaren lässt, kann diese Annahme als plausibel angesehen werden. Hätte man zu manchen Fällen (Zeilen) durch externe Verfahren oder Erkenntnisse gesicherte absolut- oder relativchronologische Datierungen, könnte man den vermuteten Zusammenhang zwischen Zeit und CA-Anordnung rechnerisch belegen. Dafür bedürfte es einer Kanonischen CA, auch als "constrained correspondence analysis" (kurz CCA) bezeichnet. Die Datierungsvariable müsste dann als kanonische Variable verwendet werden und das Ergebnis - die bedingte erste Achse der CCA - mittels Permutationstests auf Signifikanz getestet werden.

Im Zusammenhang mit der Anwendung einer CA zur Erzeugung einer relativen Chronologie wird häufig von einem sog. Parabeltest gesprochen. Einen statistischen Test dieser Art gibt es (leider noch) nicht. Man meint stattdessen, dass sich die Zeilen- und Spaltenpunkte im Raum der ersten beiden CA-Achsen in der Art einer Parabel anordnen. Da der Begriff Test in der Statistik eine genau festgelegte Bedeutung besitzt, wird empfohlen, den Ausdruck „Parabeltest“ NICHT in wissenschaftlichen Publikationen zu verwenden, da er hochgradig irreführend ist. Mit „Parabeltest“ ist ja genau genommen gemeint, dass man aufgrund des eigenen subjek-

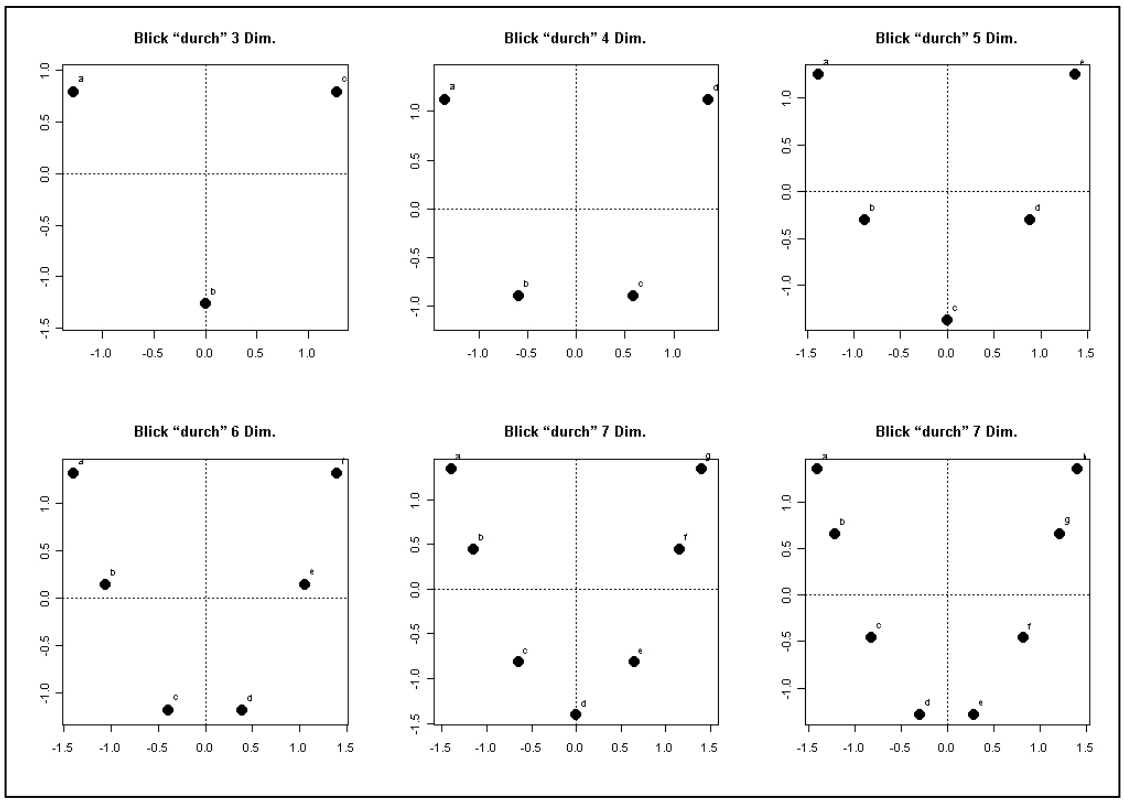
tiven visuellen Eindrucks eine parabelförmige Punkteverteilung im CA-Biplot zu erkennen meint. Dieser Begriff bezeichnet also das genaue Gegenteil eines Tests, nämlich einen individuellen Eindruck – und dann gilt „Your guess is as good as mine“. Korrekter ist es, von einer parabelförmigen Punktanordnung zu sprechen.

Der Grund für die parabelförmige Anordnung der Punkte liegt in der Dimensionsreduktion der CA (ich danke Herrn Patrick Mair/Wirtschaftsuniversität Wien für seinen Hinweis zum Verständnis der Parabel). Man stelle sich ein dreidimensionales Koordinatensystem vor. Auf jeder der drei Achsen liegt beim Koordinatenwert 1 jeweils ein Punkt. Die Koordinaten der drei Punkte lauten also:

	x	y	z
P1	1	0	0
P2	0	1	0
P3	0	0	1



Es lässt sich nun eine Ebene zwischen den drei Punkten aufspannen. Blickt man genau senkrecht durch den Mittelpunkt dieser Ebene auf den Koordinatenursprung, so ergeben die drei Punkte ein gleichseitiges Dreieck in dieser Ebene. Sie existieren in drei Dimensionen, lassen sich aber in zweien Abbilden. Zu beachten ist, dass die drei Koordinatenachsen alle drei senkrecht zueinander stehen – mathematisch ausgedrückt, sie sind orthogonal zueinander. Wenn – was ich mir z. B. nicht visuell vorstellen kann – vier Punkte jeweils beim Koordinatenwert 1 auf vier zueinander orthogonalen Achsen liegen lässt sich ein dreidimensionaler Körper finden, an dessen Ecken die Punkte liegen. Beim senkrechten Blick in 2D durch diesen Körper scheinen die Punkte an den Ecken eines gleichschenkligen Trapezes zu liegen.



Wenn man nach diesem Prinzip immer mehr Dimensionen hinzunimmt und das Ergebnis jeweils in 2D abbildet, entsteht eine Anordnung, die mit jeder neuen Dimension immer mehr die Form einer Parabel annimmt. Ein R-Code zur Visualisierung findet sich unter 12.8.

Die Parabel (bei einer Chronologieanwendung) der CA entsteht nun dadurch, dass die einzelnen Ausprägungen des Spaltenmerkmals gleichmäßig entlang eines einzigen (!) Gradienten versetzt zueinander zu und dann wieder abnehmen. Im Fall der Chronologieanwendung z. B. hat eine Ausprägung einen Teil ihrer Zu- und Abnahme gemein mit „zeitlich" benachbarten Ausprägungen, ein Teil aber ist von diesen unabhängig. Unabhängigkeit ist in einem Koordinatensystem durch Orthogonalität repräsentiert. Die einzelnen Ausprägungen könnte man also auch als ein (multidimensionales) orthogonales Achsensystem ansehen. Und der dimensionsreduzierende „Blick" der CA durch diesen multidimensionalen Raum erzeugt den Eindruck (!), die Spaltenpunkte lägen (!) auf einer Parabel. Gäbe es mehr als einen Gradienten, so würden Zu- und Abnahmetendenzen im multivariaten Raum quasi in verschiedene Richtungen weisen. Orthogonalität - und damit die Parabel - könnte bei mehr als einem Gradienten also nur dann entstehen, wenn die einzelnen Gradienten völlig voneinander unabhängig sind. Und dies dürfte in der Realität kaum auftreten. Eine leidlich parabelförmige Verteilung ist deshalb ein guter Hinweis auf die Präsenz nur eines Gradienten.

Für die CA-Spezialisten unter den Statistikern ist die Parabel einfach ein rechnerisches Artefakt – ihre Existenz beruht ja auf der Berechnungs- und Darstellungsweise der CA und stellt kein reales Phänomen dar. Deshalb interessieren sich die Spezialisten vor allem an Methoden zur Beseitigung der Parabel – etwa der „detrended CA". Angesichts des Fortschritts der matrixalgebrabasierten multidimensionalen Statistik sollte es zwar möglich sein, zu schätzen, wie eine parabelförmige CA-Punktverteilung bei bestimmten Daten „optimalerweise" aussehen sollte, wie nahe ihr die empirisch vorliegende CA dieser Daten tatsächlich kommt, und wie wahrscheinlich diese Situation durch Zufallseinflüsse verursacht sein könnte. Allein den Statistikern – und ich habe einige weltweit deswegen angefragt – fehlt das Verständnis und das Interesse für dieses Problem der Archäologen…

10. „Seriation"

Schön und gut, eine CA kann also Zeilen-/Spalten-Punkte so in einem dimensionsreduzierten Raum anordnen, dass die einander ähnlichen Zeilen/Spalten nahe beieinander zu liegen kommen und gleichzeitig möglichst viel der Ausgangsinformation zu (Un-)Ähnlichkeit der Zeilen und Spalten auf wenigen Achsen abgebildet wird. Aber wie kann denn die CA zur Diagonalisierung einer Kreuztabelle – dem was umgangssprachlich als Seriation bezeichnet wird – beitragen? Nun, am Anfang der CA stand die Kreuztabelle der zwei Nominalvariablen. Ordnet man die Zeilen und Spalten dieser Tabelle jetzt neu an, und zwar jeweils in einer Reihenfolge, die der Reihenfolge ihrer Koordinatenwerte (auf der ersten CA-Achse) entspricht, ergibt sich eine sog. diagonalisierte Tabelle. Also ganz oben käme etwa die Zeile mit der niedrigsten Koordinate auf der ersten CA-Achse, dann die Zeile mit der zweitniedrigsten usw. Gleichermaßen verführe man mit den Spalten. Da es nur um die Reihenfolge geht, ist hier die Koordinatenskalierung bedeutungslos – die Multiplikation mit den Singulärwerten verändert ja nur die Werte und nicht deren Reihenfolge.

Nach dieser Zeilen-/Spaltenneuanordnung ist zu beobachten, dass entlang der Diagonalen die Tabellenzellen mit hohen Werten liegen, abseits davon die Zellen mit niedrigen Werten. Der Grund dafür ist ganz einfach nachvollziehbar: Bei einer Anordnung nach ihren CA-

Koordinaten kommen beispielsweise die zwei sehr ähnlich zusammengesetzte Zeilen D und E nebeneinander zu liegen. Eine weitere Zeile A mag zu D ähnlicher sein als zu E. Man wird also A oberhalb und E unterhalb von D anordnen. Verfährt man so mit allen Zeilen und gleichermaßen mit den Spalten, wird man nach kurzer Zeit beobachten können, dass die neu geordnete Tabelle jetzt entlang (oder nahe bei) der Diagonalen die höchsten Zellwerte aufweist. Schließlich kommen die einander am unähnlichsten ausfallenden Zeilen (und Spalten) an den entgegengesetzten Enden der Tabelle zu liegen. Deshalb lässt sich mit einer CA eine Abundanzkreuztabelle diagonalisieren.

11. Qualität der CA-Ordination

In der Archäologie werden CA-Lösungen leider seit ihrer Einführung meist völlig ungenügend publiziert. Um das Ergebnis einer CA als Leser wissenschaftlich-kritisch einschätzen zu können, ist es **zwingend** notwendig, die Qualitätskennwerte einer CA-Lösung betrachten zu können.

Ebenso wird in der Regel vergessen, die Darstellungsart des Biplots anzugeben. Im schlimmsten Fall können CA-Lösungen nämlich zwar „schön“ aussehen, aber wissenschaftlich sinnlos, ihre Abbildung nicht nachvollziehbar und ihre Interpretation falsch sein. Die Publikation einer CA-Lösung ohne Qualitätskennwerte und Information zur Abbildungsweise kann daher nicht als gute wissenschaftliche Praxis bezeichnet werden. Ursache für diese Fehlentwicklung dürfte die im Fach übliche Vorgehensweise sein, das Herangehen anderer nur immer zu reproduzieren, ohne ein eigenes Verständnis für die Technik des angewendeten Verfahrens zu entwickeln.

Da Kritik immer konstruktiv sein sollte, gilt es im Folgenden, diese Qualitätskennwerte zu beschreiben.

Die Messung der Bedeutung einer Achse anhand des auf dieser Achse erfassten Streuungsanteils wurde oben bereits beschrieben. Es bleibt nur anzumerken, dass bei einem zweidimensionalen Biplot natürlich nur zwei Achsen – in der Regel die ersten beiden – abgebildet werden. Es ist also wichtig zu wissen, wie viel Prozent der Gesamtinertia die erste und wie viel die zweite Achse erfasst hat. So kann man beurteilen, wie viel Prozent der Gesamtinformation, die in der Ausgangskreuztabelle steckte, in der vorliegenden Grafik noch vorhanden ist. Neben den entsprechenden Inertiawerten wird häufig ein Balkendiagramm, ein sog. ‚screeplot’, abgebildet, das die Inertia pro Achse als Säulendiagramm zeigt.

Als nächstes kann man sich der Abbildungsqualität der Zeilen und Spalten zuwenden. Wie die Masse einer Zeile die CA-Lösung beeinflusst, wurde ebenfalls schon beschrieben. Wenn man nun beispielsweise eine Lösung beurteilen möchte, wird man zuerst die weiteren Kennwerte der massereichen Zeilen-/Spalten-Punkte betrachten. Diese Punkte sind wie gesagt wichtig für die Lösung – aber wie gut sind sie dargestellt?

Hier kann man auf zwei Arten vorgehen. Die erste betrifft das „Zustandekommen“ der Achsen. Die zweite die Abbildungsqualität der Punkte. Zuvor sollte man jedoch noch eine allgemeine Information betrachten, nämlich, wie viel Inertia besitzen eigentlich die einzelnen Zeilen und Spalten. Eine Zeile kann sich ja stark von der Durchschnittszeile unterscheiden, dann kann sie eine größere Inertia besitzen, oder sie unterscheidet sich kaum, dann wird sie in der Regel auch kaum Inertia besitzen. Ihre Inertia ist das Produkt aus der Chiquadratdistanz zur

Durchschnittszeile und ihrer Masse. Am Kennwert Zeilen-/Spalten-Inertia erkennt man also, welche dieser Kategorien sich besonders stark von der durchschnittlichen Zeilen-/Spalten-Zusammensetzung unterscheidet und wie stark oder schwach Zeilen und Spalten differieren.

Die Achsen erfassen nun diese Inertia. Aber welche Zeilen/Spalten haben mit ihrer Inertia zu der Inertia beigetragen, die auf einer bestimmten Achse erscheint bzw. von ihr erfasst wird? Bildlicher gesagt, welche Zeilen/Spalten haben diese Achse mit ihrer Inertia „aufgespannt"? Um die Zeilen/Spalten miteinander zu vergleichen, ist es sinnvoll, hier keine Absolutbeträge an Inertia anzugeben, sondern den Anteil in Prozent, zu dem die Inertia einer Achse auf eine Zeilen-/Spalten-Inertia zurückgeht. Man kann so einschätzen, zu welchem Anteil eine Achse auf der (Un-)Ähnlichkeit welcher Zeilen/Spalten basiert. Dies wird einfach als „contribution" oder häufiger als „absolute contribution" bezeichnet. Es geht hier also um den Inertia-„Beitrag" einer Zeile/Spalte zur Inertia einer Achse, oder anders gesagt, welchen Unterschied zwischen welchen Zeilen bzw. Spalten misst eine Achse eigentlich und als was kann sie daher interpretiert werden.

Noch interessanter als das Zustandekommen der Achsen ist die Darstellungsqualität der Zeilen-/Spalten-Punkte auf den ersten beiden Achsen. Jetzt wird einfach der Spieß umgedreht: Anstatt zu fragen, welchen Beitrag eine Zeile/Spalte zur Achseninertia leistet, fragt man nun, welcher Anteil der Zeilen/Spalten-Inertia von einer Achse erfasst wird. Man könnte auch sagen, wie viel Prozent der (Un-)Ähnlichkeit einer Zeile oder Spalte ist auf dieser Achse erfasst, oder anders gesagt, wie gut wird die Information zu dieser Zeile oder Spalte von einer Achse repräsentiert. Je höher dieser Wert ist, desto mehr Information, die diese Zeile/Spalte enthält, wird von der Achse erfasst – etwas vereinfacht ausgedrückt, desto besser ist die Zeile/Spalte auf der Achse abgebildet. Dieser Kennwert wird als „relative contribution" oder „correlation" bezeichnet. Man kann ihn als Korrelation, also Übereinstimmung, von Zeilen/Spalten-Inertia und der von der Achse repräsentierten Inertia verstehen. Bildet man das CA-Ergebnis mit den ersten beiden CA-Achsen ab, enthält diese Darstellung den höchsten Informationsgehalt, den man mit einer zweidimensionalen Darstellung erreichen kann. Die Summe der „relative contribution" für die ersten beiden Achsen, also der Anteil der Zeilen/Spalten-Inertia, der von diesen ersten beiden Dimensionen erfasst wird, ist also die Darstellungsqualität der Zeile/Spalte und wird kurz als „quality" bezeichnet. Ein Zeilen/Spalten-Punkt mit geringer Darstellungsqualität wird mit seiner Lage im Raum der (Un-)Ähnlichkeit von den ersten beiden Achsen nicht gut abgebildet, seine Unterschiede zu den anderen erscheinen in dieser Abbildung also nicht so gut. Er liegt (vielleicht) scheinbar zu nah (oder zu weit) von den ihm (un-)ähnlichen anderen Punkten entfernt. Seine Lage sollte nicht detailliert interpretiert werden.

12. CA in R mit dem Paket ‚ca'

Das verständlichste und am besten mit Literatur dokumentierte CA-Paket in R wird von einem der weltweit führenden CA-Experten, Michael Greenacre (Universität Barcelona), mitbetreut, der zur Anwendung der CA auch einen eigenen Internetauftritt präsentiert (http://carmen.org/). Das Paket heißt ‚ca' und ist via CRAN verfügbar. Als Einführungsliteratur zur CA ebenso wie als Begleitbuch zum Paket ‚ca' ist folgendes Buch zu empfehlen:
M. Greenacre, Correspondence Analysis in Practice (Boca Raton 2007).
Die hier referierten Inhalte entstammen weitgehend diesem Buch.

Zuerst wird das Paket installiert und geladen.

```
install.packages(„ca“)
require(ca)
```

Welche Pakete man geladen hat, zeigt einem die Funktion ‚search()’:

```
search()
```

Nun sollte man noch ein Arbeitsverzeichnis erstellen und auswählen. Wenn man später Dateien aus R exportiert, dann erscheinen sie in diesem Verzeichnis. Mit ‚dir.create(“[Laufwerk] :/ [Verzeichnisname]“)’ wird ein neues Verzeichnis erstellt. Man beachte den nach rechts geneigten Querstrich.

```
dir.create("C:/R_daten")
```

Mit ‚setwd(“[Verzeichnisbaum]“)’ wird das Verzeichnis angewählt.

```
setwd("C:/R_daten")
```

Mit ‚getwd()’, das den Namen des aktuellen Arbeitsverzeichnisses ausgibt, wird der Erfolg der Aktion überprüft.

```
getwd()
```

12.1. Beispieldatenerzeugung

Für die folgende CA wird ein Beispieldatensatz aus Zufallsdaten erzeugt, wobei diese Zufallsdaten eine klare Gruppenstruktur aufweisen werden. Für eigene Experimente mit einer neuen Methode empfiehlt es sich, entweder einen gut bekannten Datensatz zu verwenden, oder einen mit einer genau bekannten Struktur zu erzeugen.

Zunächst wird eine leere Matrix erzeugt. Eine Matrix ist ein Zahlenfeld. Es ähnelt einer Datentabelle, nur dass alle Einträge einer Matrix von der gleichen Art sein müssen, also nur Zahlen oder nur Buchstaben. Das erste Argument bezeichnet die Matrixeinträge: hier soll in allen Zellen eine Null stehen. Das zweite legt die Anzahl der Zeilen und das dritte die Anzahl der Spalten fest.

```
matrix(0,30,6)->m
```

Bevor man Zufallszahlen zieht empfiehlt sich das Setzen des Zufallszahlengenerators. Auf diese Weise erhält jeder, der die gleiche Setzung vornimmt, auch die gleichen Zahlen.

```
set.seed(100)
```

Die folgende programmierte Schleife läuft zehnmal durch. Jedes Mal erzeugt sie einen Vektor, der aus drei Teilen zusammengesetzt ist und schreibt in die i-te Zeile der gerade erzeugten Matrix. Die drei Teile des Vektors werden jeweils auf die gleiche Weise erzeugt. Es werden je zwei Zahlen einer poissonverteilten Zufallsvariablen gezogen, wobei der Erwartungswert für die ersten beiden Ziehungen 8, der für die zweiten beiden 3 und der letzte 1 beträgt. Auf diese Weise entstehen Zeilen, die auf ihren ersten beiden Plätzen sehr hohe Zahlen aufweisen, auf ihren zweiten beiden mittelhohe Zahlen und auf ihren beiden letzten Plätzen niedrigen Zahlen.

```
for (i in 1:10)
  { m[i,]<-c(rpois(2,8),rpois(2,3),rpois(2,1))  }
```

In die Zeilen 11 bis 20 der Ausgangsmatrix werden Zeilen geschrieben, die auf ihren mittleren beiden Positionen hohe Zahlen enthalten.

```
for (i in 11:20)
 { m[i,]<-c(rpois(2,2),rpois(2,8),rpois(2,2))  }
```

Die Zeilen 21 bis 30 der Ausgangsmatrix werden mit Zeilen beschrieben, die auf ihren letzten beiden Positionen hohe Zahlen enthalten, auf ihren mittleren beiden mittelhohe und auf ihren ersten beiden niedrige.

```
for (i in 21:30)
 { m[i,]<-c(rpois(2,1),rpois(2,3),rpois(2,8))  }
```

Diese Matrix soll also eine beliebige Abundanzenkreuztabelle darstellen mit 30 Fällen in den Zeilen und den Typen eins bis sechs in den Spalten. In ihren ersten zehn Zeilen enthält sie eine Gruppe von Fällen, die durch besonders hohe Anzahlen der Typen 1 und 2 charakterisiert ist. Die Fälle 11 bis 20 beinhalten viele Typen 3 und 4. Schließlich treten die Typen 5 und 6 vor allem in den Fällen 21 bis 30 auf. Es gibt also drei klare Gruppen, die Zeilen 1 bis 10, 11 bis 20 und 21 bis 30.

Als nächstes wird eine Nominalvariable erzeugt, die diese Information enthält: jeweils zehnmal die gleiche Gruppenzugehörigkeit. In R ist dies ein Vektor der Klasse ‚factor'.

```
  factor (sort(rep(1:3,10)), labels=c("gruppe1", "gruppe2", "gruppe3") ) ->
gru
  gru
  is.factor(gru)
```

Als nächstes wird eine Datentabelle aus der Abundanzentabelle und dem Gruppenzugehörigkeitsmerkmal erzeugt.

```
  data.frame(m, gruppe= gru) -> da
```

Dann erhalten die Abundanzen noch ordentliche Namen. Mit dem Befehl ‚paste' werden jeweils die einander entsprechenden Einträge aus zwei Vektoren mit einander zu Einträgen eines Vektors „verklebt". Hier enthält der erste Vektor nur den Eintrag „Typ". Der zweite Vektor ‚1:ncol(m)' besteht aus einer Zahlenreihe von 1 bis zu der Zahl, die der Spaltenanzahl der Ausgangsmatrix entspricht. Getrennt werden sollen die beiden jeweils durch „kein Freizeichen" – dies legt das Argument ‚sep="" ' fest.

```
  paste("Typ",sep="", 1:ncol(m) ) -> colnames(da)[1:6]
```

Die Zeilen der Datentabelle erhalten auf die gleiche Weise Namen:

```
  paste("Fall",1:nrow(m))  -> rownames(da)
```

Den Aufbau der fertigen Datentabelle zeigt der Befehl ‚str()'.

```
  str(da)
```

Die ersten 4 Zeilen der Datentabelle gibt der Befehl ‚head()' aus, wobei das Argument die Anzahl der sichtbaren Zeilen angibt.

```
head(da, 4)
```

12.2. Berechnung

Die eigentliche Korrespondenzanalyse führt der kleine Befehl ‚ca()' durch. Er bedarf als Argument einer Matrix (mit ganzen Zahlen), der Spalten und Zeilen Namen tragen, oder der Spalten einer Datentabelle – ein Objekt der Klasse „data.frame" – die Abundanzen enthalten oder einer Kreuztabelle – ein Objekt der Klasse „table", das mit dem gleichnamigen Befehl erzeugt wird. Hier wird die CA auf die ersten sechs Spalten der Datentabelle ‚da' angewendet, da in ihnen die oben erzeugten Beispielabundanzen stehen.

```
ca(da[,1:6]) -> cada
```

Das war's, die CA ist fertig. Als erstes betrachtet man das CA-Objekt, indem man einfach seinen Namen eintippt.

```
cada
```

Man erhält eine umfangreiche Ausgabe. Die erste kleine Tabelle namens ‚Principal inertias' enthält für jede CA-Achse in der Zeile ‚value' die Inertia, die eine Achse erfasst und in der Zeile ‚Percentage' den entsprechenden Anteil an der Gesamtinertia. In der schon unübersichtlicheren zweiten Tabelle namens ‚Rows' werden Informationen zu den Zeilen geboten. Jeweils unter dem Namen einer Zeile steht ihre Masse, ihr Chiquadratabstand zum Achsennullpunkt, ihre Inertia sowie ihre Koordinate auf der ersten (‚Dim. 1') und zweiten Achse (‚Dim. 2'). In der dritten Teiltabelle namens ‚Columns' wiederholt sich das ganze für die Spalten.

Der oben angewendete Befehl ‚ca()' erzeugt ein spezifisches CA-Objekt. Den Aufbau dieses Objektes zeigt wiederum der Befehl ‚str()'.

```
str(cada)
```

Das CA-Objekt gehört zur Klasse „list", ist also eine Liste. Der erste Listeneintrag ist ein numerischer Vektor mit den Singulärwerten. Der zweite Listeneintrag ist nicht von Interesse. Der dritte Listeneintrag namens ‚$rownames' enthält die Zeilennamen der Abundanzenkreuztabelle, also unsere Fallnamen, als Vektor mit Zeicheneinträgen (‚chr' für „character"). Der Eintrag ‚$rowmass' enthält als numerischer Vektor die Zeilenmassen. ‚$rowdist' ist der numerische Vektor mit den Chiquadratabständen zum Achsennullpunkt, ‚$rowinertia' ist der Vektor mit den Inertiabeträgen der Zeilen und ‚$ rowcoord' schließlich ist eine Matrix. Sie enthält die Zeilenkoordinaten der CA-Lösung: in der ersten Spalte stehen die Koordinaten auf der ersten Achse usw. Zeilen- wie Spaltenkoordinaten sind hier im Standardformat angegeben! ‚$rowsup' wird später noch einmal wichtig; hier nur soviel, es kann die Nummern sog. Ergänzungszeilen enthalten. Die Einträge mit dem Präfix ‚col-' enthalten die entsprechenden Informationen zu den Spalten.

Jeden Eintrag dieser Liste kann man mit seinem Namen einzeln ansprechen bzw. extrahieren.

```
cada$rowcoord -> rowstd
```

So extrahiert man z. B. die Matrix mit den Zeilenstandardkoordinaten.

12.3. Ergänzungszeilen/-spalten

Oben wurde auf das Problem von Ausreißerzeilen/-spalten und eine neue Lösung dafür hingewiesen, die Ergänzungszeilen/-spalten. Solche Ausreißerzeilen/-spalten besitzen meist extrem hohe Inertia und zwingen die Achsen daher, zunächst ihre Inertia zu erfassen. Und so „überdeckt“ ihre Inertia die der restlichen Zeilen/Spalten. Dadurch geprägte CA-Ergebnisse verfehlen meist ihr Ziel, eine Anordnung der Zeilen/Spalten unter Berücksichtigung der eigentlichen Dateninformationen zu erreichen.

Will man Spalten als Ergänzungsspalten auszeichnen, setzt man das Argument ‚supcol=’. Als Parameter für das Argument werden einfach die Nummern der entsprechenden Spalten aufgeführt. Die Funktion ‚c([Eintrag1], [Eintrag2], [Eintrag3])’ verbindet dabei alle von Kommata getrennten Einträge zu einem Vektor. Wollte man nur eine Spalte ausschließen, könnte man deren Nummer anstatt der Funktion ‚c()’ angeben. Im unten folgenden Beispiel werden die Spalten 1 und 3 ausgeschlossen. Die folgenden beiden Befehlszeilen enthalten zugleich bereits einen Befehl zur Grafikerzeugung, um die Ergebnisse vergleichen zu können. Die Ergänzungssparten werden als ungefüllte Punkte gezeichnet.

Ein CA-Biplot:

```
plot(ca(da[,1:6], map="rowprincipal")
```

Der Biplot einer CA mit den Spalten 1 und 3 als Ergänzungsspalten:

```
plot(ca(da[,1:6], supcol=c(1,3)), map="rowprincipal")
```

Zeilen deklariert man mit dem Argument ‚suprow=’ zu Ergänzungszeilen. Mehrere einander folgende Zeilen kann man etwa durch die Ansprache mit einer Zahlenreihe (Vektor) auswählen. Die Angabe ‚1:5’ erzeugt einen Vektor, der die Zahlen von 1 bis 5 enthält.

```
plot(ca(da[,1:6], suprow=c(1:5)), map="rowprincipal")
```

12.4. Qualitätskennwertausgabe im Paket ‚ca’

Die wichtigen Qualitätskennwerte der CA-Lösung erhält man mit dem Befehl ‚summary()’, angewendet auf das CA-Objekt.

```
summary(cada)
```

Man erhält wieder eine Liste mit drei Einträgen. Der erste ist eine kleine Tabelle. Unter ‚dim’ steht die Nummer der CA-Achse, also erste, zweite usw. Unter ‚value’ steht wiederum die von der Achse erfasste Inertia. In der Spalte ‚%’ ist der Anteil der Achseninertia an der Gesamtinertia aufgeführt und die Spalte ‚cum%’ enthält in der n-ten Zeile die Summe der Inertia, die die Achsen 1 bis n insgesamt erfassen. Wichtig ist hier, wieviel Prozent Achse 1 und 2

zusammen abbilden. Die Spalte ‚scree plot' ist nur eine einfache graphische Umsetzung der Information der Spalte ‚cum%'.

Die nächste Tabelle namens ‚Rows' enthält die Qualitätskennwerte für die Zeilen. In der ersten Spalte steht der Name, in der zweiten die Masse. Die Spalte ‚qlt' enthält die Abbildungsqualität auf den ersten beiden Achsen in Promille, ein Wert von 999 bedeutet also sehr gute, ein Wert von 13 eine sehr schlechte Abbildungsqualität. Unter ‚inr' ist wie zu erwarten die Inertia aufgeführt und zwar als Anteil an der Gesamtinertia in Promille. Die Spalten ‚k=1' und ‚k=2' enthalten die Koordinaten auf der ersten bzw. zweiten Achse wiederum in Promille. Es folgt jeweils eine Spalte namens ‚cor' und eine namens ‚ctr' mit den Kennwerten für die entsprechende Achse. Unter ‚cor' ist in Promille die Korrelation („relative contribution") zwischen Zeile und Achse, also der Anteil der Zeileninertia angegeben, der von dieser Achse erfasst wird. Die Summe der beiden ‚cor'-Werte ergibt den Wert in der Spalte ‚qlt'. Unter ‚qtr' steht der Beitrag („absolute contribution") an Inertia in Promille, den die Zeile zur jeweiligen Achse leistete. Die ‚cor'-Werte zeigen einem also die Abbildungsqualität (s. o.) und die ‚ctr'-Werte informieren darüber, was die jeweilige Achse misst.

Das Ganze wiederholt sich schließlich für die Spalten mit der Tabelle ‚Columns'.

Wiederum lässt sich jeder Eintrag dieser Liste mit seinem Namen einzeln ansprechen bzw. extrahieren. Allerdings ist die konkrete Ansprache der einzelnen Spalten, etwa der Zeilenqualitätskennwerttabelle, durch eine kleine programmiererische Inkonsistenz etwas erschert.

```
summary(cada)$rows
```

Dieser Befehl ergibt die Kennwerttabelle. Aber da die Spaltennamen dieser Tabelle dummerweise Leerzeichen und Dubletten enthalten, wie der folgende Befehl zeigt, lassen sich deren Spalten nur über einen Umweg ansprechen.

```
names(summary(cada)$rows)
```

Am besten man spricht die interessierende Spalte mit ihrer Indexzahl an.

```
summary(cada)$rows[ ,3]
```

Die Zahlen in eckigen Klammern sprechen bei einem zweidimensionalen R-Objekt, also etwa einer Datentabelle („data.frame") oder einer Matrix die Zeilen und Spalten dieses Objektes an. Dabei bedeutet ein Leerzeichen die Ansprache aller Einträge dieser Dimension. Die Zahl vor dem Komma spricht die Zeilennummer an, die dahinter die Spaltennummer. Die hier benutzte Ansprache wählt also alle Zeilen aus – Leerzeichen vor dem Komma – und bei allen Zeilen jeweils die dritte Spalte – die ‚3' hinter dem Komma. In diesem Fall entspricht die dritte Spalte der Kennwerttabelle der Spalte namens ‚qlt', also der Darstellungsqualität. Wollte man nur die Darstellungsqualität der Fälle 11 bis 20, so würde der Befehl lauten:

```
summary(cada)$rows[ 11:20 , 3]
```

Die beiden Inertiabeiträge zu den Achsen ergäbe der Befehl:

```
summary(cada)$rows[ , c(7, 10) ]
```

Um die Qualitätskennwerte bei der CA-Publikation im Anhang anzugeben, muss man sie exportieren. Zunächst wird das „Summary-Objekt" erzeugt.

```
summary(cada) -> sucada
```

Dann wird ein Vektor mit eindeutigen (!) Spaltennamen für die zu extrahierenden Datentabellen erstellt.

```
c("Name","Masse","Qual","Inertia","Korr_1","Beitr_1","Korr_2","Beitr_2")->nama
```

Schließlich werden aus der Datentabelle mit den Qualitätskennwerten zu den CA-Zeilenpunkten die Spalten mit den Einträgen zu Name, Masse, Abbildungsqualität, Inertia, Korrelation mit Achse 1, Beitrag zu Achse 1 sowie Korrelation mit Achse 2 und Beitrag zu Achse 2 in ein eigenes R-Objekt geschrieben.

```
(sucada$rows[,c(1:4,6:7,9:10)]->zeiqual)
```

Das Gleiche wird für die CA-Spaltenpunkte wiederholt.

```
(sucada$columns[,c(1:4,6:7,9:10)]->spaqual)
```

Jetzt erhalten die Spalten der beiden R-Objekte noch die neuen Namen.

```
names(zeiqual)<-nama
names(spaqual)<-nama
```

Mit dem Befehl ‚write.csv2()' wird eine Datentabelle („data.frame") exportiert und als Datei im .csv-Format auf ein Speichermedium geschrieben. Beim .csv-Format liegt die exportierte Datei im ASCI-Zeichensatz vor, wobei die Spalteneinträge einer Datenbankzeile durch ein Semikolon getrennt werden. Eine derartige Datei kann mit jedem Texteditor oder auch EXCEL geöffnet werden.

Der Befehl ‚write.csv2()' nimmt als erstes Argument den Namen der zu exportierenden Datentabelle. Das zweite Argument ist der Name, den die Datei erhalten soll. Das nächste Argument ‚row.names=' schreibt mit der Setzung „TRUE" die Zeilennamen der R-Datentabelle mit in die exportierte Datei, bei „FALSE" lässt es sie weg. Das Argument ‚quote=' legt fest, ob die von R als Texteintrag behandelten Partien in der späteren Datei in Anführungsstrichen stehen oder nicht.

```
write.csv2(zeiqual, "Zeilenqualität.csv", quote=FALSE, row.names=FALSE)
write.csv2(spaqual, "Spaltenqualität.csv", quote=FALSE, row.names=FALSE)
```

Mit dem Befehl ‚dir()', der den Inhalt des aktuellen Arbeitsverzeichnisses ausgibt, wird der Erfolg der Aktion überprüft.

```
dir()
```

Die Zeilenkoordinaten in Prinzipalskalierung erhält man durch die Multiplikation – das Zeichen %*% steht für Matrixmultiplikation – der Zeilenstandardkoordinaten mit den Singulärwerten, wobei diese die Diagonaleinträge einer ansonsten nur Nullen enthaltenden Matrix bilden müssen. Dies bewirkt der Befehl ‚diag()', der als Argument des Vektors für die Diagonaleinträge bedarf.

```
cada$rowcoord %*% diag(cada$sv) -> rowprc
```

Ergebnis ist eine Matrix mit den entsprechenden Einträgen. Für den Export erhält sie jetzt noch die Namen der Fälle als Zeilennamen.

```
(cada$rownames->rownames(rowprc))
```

Auch die Spalten der Koordinatenmatrix erhalten auf die bekannte Art Namen. Dabei erzeugt „1:ncol(rowprc)“ einen Vektor, der die Zahlen von 1 bis zur Anzahl der Spalten von „rowprc“ enthält.

```
paste("Achse", sep="", 1:ncol(rowprc))->namo
```

Der Namensvektor wird zugewiesen.

```
(namo->colnames(rowprc))
```

Schließlich wird das Ganze noch in eine Datei geschrieben.

```
write.csv2(rowprc, "ZeilPrinkoord.csv", quote=FALSE, row.names=TRUE)
```

12.5. CA-Biplot (Achsenkreuzausdruck)

Mit R kann man aber nicht nur eine CA mit allen wichtigen Kennwerten berechnen. R erlaubt außerdem die Erstellung publikationsreifer Biplots:

```
plot(cada, map="rowprincipal", mass=c(TRUE,TRUE), contrib=c("relative",
"relative"), xlim=c(-2,2), ylim=c(-2,2), col=c(1,4), labels=c(0,2))
```

Das Argument ‘map=’ steuert die Koordinatenskalierung; hier werden die Zeilen in Prinzipalkoordinaten und die Spalten also in Standardkoordinaten abgebildet. Mit ‚mass =c(,)’ wird mittels TRUE/FALSE für die Zeilen und die Spalten angegeben, ob die Größe der Punktsymbole der Masse entsprechen soll. Mit dem Argument ‚contrib =c(,)’ wird für Zeilen und Spalten mit den Parametern „relative“ oder „absolute“ angegeben, ob die Farbintensität eines Punktes den Inertiabeitrag zu den Achsen („absolute“) oder die Darstellungsqualität („relative“) repräsentieren soll. Die Argumente ‚xlim’ und ‚ylim’ steuern die Unter- und Obergrenze der Grafik in CA-Koordinaten. Mit ‚col=c(,)’ werden die Farben für Zeilen und Spalten gewählt. Das Argument ‚labels= c(,)’ legt für Zeilen und Spalten fest, ob nur die Punkte (0), nur die Namen (1) oder Punkte und Namen (2) dargestellt werden sollen.

Diese Abbildung hat noch keine Beschriftung. Das lässt sich mit dem Sekundär-Grafik-Befehl ‚title()’ nachholen. Die Argumente ‚xlab=“ “ ’ und ‚ylab=“ “ ’ erhalten die Bezeichnungen für die X- und die Y-Achse der Grafik. Mit ‚cex.lab=’ wird ein Zoomfaktor für den Achsentiteltext eingestellt – hier das 0,9-fache der normalen Schriftgröße. Mit ‚main=““ ’ wird die Grafiküberschrift erzeugt, mit ‚sub=““ ’ die Grafikunterschrift, wobei ‚cex.sub=’ wiederum den Verkleinerungsfaktor für diesen Texteintrag steuert. Dabei müssen die Texteinträge stets in Anführungsstrichen stehen, da R sonst nach einem Vektor dieses Namens sucht.

```
title(xlab="erste CA-Hauptachse", ylab="zweite CA-Hauptachse",
cex.lab=.9, main="Biplot einer Korrespondenzanalyse", sub="(symmetrischer
Biplot mit Zeilen in Prinzipalkoordinaten)", cex.sub=.75,)
```

Der einfachste Weg, diese Ergebnisgrafik zu exportieren, läuft über das Menü. Dafür klickt man ins Grafikfenster und dann auf Datei. Jetzt wählt man *"Speichern als"*, dann *"jpeg"* und schließlich *"100 % Qualität ..."* und muss nur noch den Ordner wählen sowie den Dateinamen eingeben.

Wie gut hat die CA eigentlich die den Beispieldaten „eingebaute“ Gruppenzugehörigkeit der Fälle abgebildet? Diese Zugehörigkeit ist im Merkmal ‚da$gruppe’ gespeichert. Sie soll nun in eine CA-Grafik einfließen. Mit der folgenden Grafik wird zunächst beim Ausdruck des CA-Biplots die Darstellung der Punkte durch das Argument ‚what=c("none", "none")’ unterdrückt.

```
plot(cada, map="rowprincipal", xlim=c(-2,2), ylim=c(-2,2), what =
c("none","none") )
```

Die Zeilenpunkte werden jetzt als Punkttyp 21 (‚pch=21’) mit den möglichen Füllfarben rot, gelb und blau (bg=c(2,7,4)) abgebildet. Dabei steuert der Ausdruck in Klammern ‚[da$gruppe]’ welche Farbe gewählt wird: das Merkmal ‚da$gruppe’ ist eine Nominalvariable mit drei Ausprägungen. Die Punktfarbe wird jetzt entsprechend der Ausprägung des Merkmals ‚gruppe’ gewählt; die erste Ausprägung ergibt rote Punkte, die zweite gelbe und die dritte blaue. Als Koordinatenmatrix für die Punkte wurden die Spalten eins und zwei der oben berechneten Zeilenprinzipalkoordinaten gewählt.

```
points(rowprc[,1:2], pch=21, bg=c(2,7,4)[da$gruppe])
```

Die Punkte für die Spalten (in Standardkoordinaten) zeichnet der folgende Befehl. Dabei werden dreieckige gefüllte Punkte (‚pch=24’) verwendet, deren Füllfarbe grün ist (‚bg=3’) und die 1,1-mal größer sind als die Zeilenpunkte.

```
points(cada$colcoord[,1:2], pch=24, bg=3, cex=1.1 )
```

Die Beschriftung wird mit dem Befehl ‚text()’ erzeugt. Er nimmt als erstes Argument die Koordinaten der Textelemente, als zweites den Vektor mit den Textelementen selbst – hier sind es die Spaltennamen; zuletzt werden die Textelemente gegenüber den angegebenen Koordinaten alle um 0,4 Einheiten nach rechts versetzt, um nicht auf den Punkten zu liegen zu kommen.

```
text(cada$colcoord[,1:2], cada$colnames, adj=-.4 )
```

Jetzt kommt noch eine ordentliche Beschriftung dazu:

```
title(xlab="erste CA-Hauptachse (50,6 % Gesamtinertia)", ylab="zweite CA-
Hauptachse (34,3 % Gesamtinertia)", cex.lab=.8, main="Biplot einer Kor-
respondenzanalyse von Beispieldaten", sub="Biplot mit Zeilen in Prinzi-
palkoordinaten. Die Zeilenpunkte sind nach Gruppenzugehörigkeit einge-
färbt ",  cex.sub=.7)
```

Ginge es hier um echte Daten, so wäre dies eine publikationsreife Darstellung des CA-Ergebnisses. Die Ergebnisgrafik kann wieder auf dem Weg über das Menü exportiert werden.

Um andere als die ersten beiden Hauptachsen darzustellen, muss das Argument ‚dim=’ gesetzt werden. Die erste und vierte Achse wird mit dem folgenden Befehl als Biplot ausgegeben.

```
plot(cada, map="rowprincipal", dim=c(1,4))
```

Schließlich lässt sich das CA-Ergebnis auch in einem interaktiven 3D-Biplot darstellen. Hier wurde mit dem Argument 'labels=c()' festgelegt, dass die Zeilenpunkte ohne und die Spaltenpunkte mit Beschriftung dargestellt werden.

```
plot3d(cada, map="rowprincipal", labels=c(0,2))
```

Die linke Maustaste dreht die Grafik und das Rollrädchen zoomt hinein oder heraus.

Der Export des 3D-Biplots ist allerdings nicht von hoher Qualität und kann nur im .png-Format erfolgen. Dafür gibt man als erstes Argument den Namen der zukünftigen Grafikdatei an. Die weiteren Argumente bleiben in der normalen Anwendung unverändert.

```
rgl.snapshot("cada3d.png", fmt="png", top=TRUE)
```

12.6. „Screeplot“ der Achseninertia

Die Verteilung der Streuung auf die Achsen des CA-Ergebnisses wird häufig in Anlehnung an das Eigenwertdiagramm der PCA als sog. „Screeplot“ bezeichnet. Es handelt sich um ein Säulendiagramm, bei dem die Säulenhöhe der Größe der Singulärwerte entspricht. Dafür werden die Singulärwerte aus dem einem temporären „Summary-Objekt“ extrahiert. Sie bilden dort die zweite Spalte auf der ersten Position der Liste.

```
summary(cada)[[1]][,2] -> sc
```

Mit ‚paste()’ wird ein Textvektor mit den zukünftigen Basisbeschriftungen der Säulen erzeugt. Das Argument an zweiter Stelle ‚1:length(sc)’ bildet einen Vektor, der die Zahlen von 1 bis zur Anzahl der Einträge im Vektor ‚sc’ enthält.

```
paste(”Achse”, 1:length(sc)) ->nam
```

Das Säulendiagramm gibt der Befehl ‚barplot()’ aus. Als erstes Argument wird die Datengrundlage des Diagramms angegeben. Dem Argument ‚names.arg=’ wird der Textvektor mit den Beschriftungen übergeben. Bei ‚ylim’ wird diesmal der gerundete Wert des um 20 % erhöhten größten Singulärwertes als Achsenmaximum gewählt. Mit ‚space=’ wird der Abstand zwischen den Säulen bestimmt und ‚las=1’ sorgt für eine waagrechte Beschriftung der Y-Achse.

```
barplot(sc, names.arg=nam, col=8, y-
lim=c(0,round(max(sc)+max(sc)/5,1)),space=0, las=1)
```

Dem Befehl ‚text()’ wird als X-Koordinate ein Vektor übergeben, der die Zahlen von 1 bis zur Anzahl der CA-Achsen bzw. Singulärwerte enthält, verringert um 0,5. Als Y-Koordinate werden die Singulärwertgrößen gewählt. Der Text schließlich besteht aus den auf drei Stellen gerundeten Singulärwerten.

```
text((1:length(sc))-.5, sc, round(sc,3), pos=3)
```

Schließlich wird noch durch das Argument ‚font=2’ eine Überschrift in Fettdruck erzeugt.

```
  title(main="Inertia pro Achse", font=2)
```

Die Ergebnisgrafik kann wieder auf dem Weg über das Menü exportiert werden.

12.7. Diagonalisierung („Seriation“) der Kreuztabelle

Ein weiteres wichtiges Ergebnis einer CA-Ordination ist ihre Verwendbarkeit zur Diagonalisierung der Abundanzentabelle. Wie oben unter 10. beschrieben, erfolgt die Diagonalisierung einer Kreuztabelle anhand eines CA-Ergebnisses einfach dadurch, dass sowohl die Zeilen als auch die Spalten in der Reihenfolge ihrer Koordinatenwerte auf der ersten CA-Achse angeordnet werden.

Der Befehl ‚order([Vektor])’ erzeugt selbst einen Vektor, der die Position des jeweiligen Vektoreintrages enthält, wenn der Vektor der Größe nach geordnet würde. Das ist ohne Anschauung etwas unverständlich. Zunächst wird ‚order()’ auf die erste Spalte der CA-Spalten-Koordinaten angewendet.

```
  (order(cada$colcoord[,1])->colord)
```

Das ergibt einen Vektor mit den folgenden Ganzzahlen („integer“).

```
     [1] 2 1 4 3 5 6
```

Man betrachte nun die eigentlichen Koordinatenwerte

```
   cada$colcoord[,1]
     [1] -1.11588268 -1.24818251 -0.06907465 -0.21221520  1.31588932
1.42082246
```

Würde man diese Werte der Reihe nach vom Kleinsten zum Größten ordnen, käme zuerst der zweite Wert, dann der erste, dann der vierte, dann der dritte, dann der fünfte und schließlich der sechste Wert. Das sind genau die Zahlen, die im Vektor ‚colord’ stehen. Dieser Vektor enthält also die Reihenfolge der Spalten, wenn man sie nach der Größe ihrer CA-Koordinaten anordnet.

Auf die gleiche Weise erzeugt man einen Vektor mit der Reihenfolge der Zeilen.

```
  (order(cada$rowcoord[,1])->roword)
```

Zum Verständnis der folgenden Operation wird noch einmal die Abundanzentabelle in ihrem ursprünglichen Zustand ausgegeben.

```
  da[,1:6]
```

Jetzt wird mittels der Indizierung – das ist die hier schon mehrmals benutzte Ansprache mit den eckigen Klammern – diese Tabelle geordnet: der im zweiten Klammerpaar als erster stehende Vektor der Zeilenordnung „roword“ führt dazu, dass zunächst die neunte Zeile, dann die zehnte, dann die dritte usw. ausgegeben wird. Der zweite, mit einem Komma abgesetzte Vektor spricht die Spalten an, deshalb ist hier der Vektor der Spaltenordnung „colord“ einzusetzen. Dabei ist es unerheblich, dass Zeilen- und Spaltenkoordinaten beide in ihrer Standardversion zur Ordnung benutzt wurden, schließlich geht es um die relative Reihenfolge.

```
(da[,1:6][roword,colord] -> diatab)
```

Das Objekt ‚diatab' ist die geordnete Tabelle – für R besitzt sie weiterhin die Klasse „data.frame". Mit dem Befehl ‚write.csv2()' lässt sie sich exportieren (s. o. 12.4.).

```
write.csv2(diatab, "Diagonaltabelle.csv", row.names=TRUE, quote=FALSE)
```

Die Datei „Diagonaltabelle.csv" kann jetzt mit EXCEL geöffnet und bearbeitet werden.

12.8. Nachvollziehen der Parabelentstehung

Mit dem folgenden Code kann man nachvollziehen, wie eine vermeintliche Parabel (s. o. 9.) eigentlich zustande kommt. Dafür markiere man die folgende kleine Tabelle und Drücke die Tasten „STRG" und „c". Die Tabelle liegt nun im Zwischenspeicher des PC.

```
3 1 0 0 0 0 0 0
1 3 1 0 0 0 0 0
0 1 3 1 0 0 0 0
0 0 1 3 1 0 0 0
0 0 0 1 3 1 0 0
0 0 0 0 1 3 1 0
0 0 0 0 0 1 3 1
0 0 0 0 0 0 1 3
```

Der folgende Code liest sie daraus wieder ein. Zugleich werden Kleinbuchstaben als Spaltennamen und Großbuchstaben als Zeilennamen gesetzt.

```
read.table("clipboard", col.names=letters[1:8], row.names=LETTERS[1:8]
->sida
```

Jetzt muss das Grafikfenster durch Anklicken zum aktiven Fenster gemacht werden. Das Menü sollte sich verändert haben, so dass es nur noch die Sparten „Datei", „History", „Resize" und „Windows" gibt. Unter „History" wähle man „Aufzeichnen". Dann sind der Reihe nach die folgenden Code-Zeilen abzuarbeiten.

```
plot(ca(sida[1:3,1:3]), map="rowprincipal", main="Blick “durch” 3 Dim.",
what=c("none", "all"))
plot(ca(sida[1:4,1:4]) , map="rowprincipal", main="Blick “durch” 4 Dim."
,  what=c("none", "all"))
plot(ca(sida[1:5,1:5]) , map="rowprincipal", main="Blick “durch” 5 Dim."
,  what=c("none", "all"))
plot(ca(sida[1:6,1:6]) , map="rowprincipal", main="Blick “durch” 6 Dim."
,  what=c("none", "all"))
plot(ca(sida[1:7,1:7]) , map="rowprincipal", main="Blick “durch” 7 Dim."
,  what=c("none", "all"))
plot(ca(sida[1:8,1:8]) , map="rowprincipal", main="Blick “durch” 8 Dim."
,  what=c("none", "all"))
```

Jede Grafik zeigt eine CA, bei der man "durch" mehr Dimensionen „hindurch sieht". Wen man nun das Grafikfenster anwählt, kann man mit den Tasten „Bild-auf" und „Bild-ab" zwischen den Grafiken hin und her blättern, denn durch das Anschalten der „History" verbleiben alle neue Grafiken im Zwischenspeicher. Die erste Grafik lässt sich noch leicht verstehen: man blickt senkrecht auf die Ebene zwischen den Endpunkten dreier senkrecht zueinander stehender Achsen – diese Ebene existiert in 2D. Die zweite Grafik ist der Blick „durch" einen

dreidimensionalen Körper dessen Endpunkte die Punkte auf den vier senkrecht aufeinander stehenden Achsen sind; der "Blick" ergibt in 2D ein gleichschenkliges Trapez. Nach dem gleichen Prinzip nimmt man immer mehr Achsen dazu...

12.9. Einlesen eigener Daten

Häufig liegen die beiden Nominalvariablen, auf die man eine CA anwenden will, als Spalten einer Datenbank vor. Diese Datenbank kann beispielsweise das Format .dbf oder .csv besitzen. Darin sind u. a. die interessierenden Keramikscherben und ihre Befundnummern erfasst.

Eine gängige Vorgehensweise ist es, eine Funddatenbank zu erzeugen, in der jede Zeile ein Fundstück repräsentiert und jede Spalte ein Merkmal dieses Fundes. Als Beispiel für die mögliche Struktur einer solchen eigenen Datenbank dient die Datei „cadat.csv“.

Man kann sie mit folgendem Befehl in R einlesen:

```
read.table(file.choose(), header=TRUE, sep=";")->data
```

Dabei lässt sich mit der Funktion ‚file.choose()’ die gewünschte Datei über das übliche Windows-Auswahlfenster anwählen. Zuerst wird der Aufbau überprüft.

```
str(data)
```

Das sollte folgenden Bildschirmtext ausgeben:

```
'data.frame':   160 obs. of  3 variables:
$ Bef.Nr: Factor w/ 10 levels "1","2","3","4",..: 10 2 4 2 9 8 9 7 2 1 ...
$ Typ   : Factor w/ 10 levels "A 1","A 2","A 3",..: 8 7 7 7 4 7 6 6 6 1 ...
$ Mat   : Factor w/ 2 levels "stein","ton": 2 2 2 2 2 2 2 2 2 2 ...
```

Die Nummer des Befundes, aus dem der Fund stammt, steht in der Spalte namens „Bef.Nr.“. Hier können so problematische Kürzel stehen wie z. B. der Eintrag „9 a“ für einen nachträglich in Befund 9 und 9 a geteilten Befund, der ursprünglich die Nummer 9 hatte. In der Spalte Befund stehen also Zeichen keine Zahlen, denn der Eintrag „9 a“ ist keine Zahl. Diese Zeichen sind die Ausprägungen der Nominalvariable „Befundnummer“ – man beachte, trotz der Nummerierung ist die Befundnummer eine Nominalvariable, nur dass man in der Regel Zahlen anstatt von Wörtern als Namen vergibt. R hat beim Einlesen automatisch erkannt, dass die Spalte „Bef.Nr.“ eine Nominalvariable ist, da hier ein Eintrag wie „9 a“ auftaucht, und daher die Spalte als Nominalvariable („Factor“) mit 10 Ausprägungen („ w/ 10 levels“) erfasst. Zusätzlich werden noch einige Ausprägungen (“1“, “2“, “3“ , “4“, …:) und die ersten Zelleneinträge (10, 2, 4, 2 usw.) aufgeführt.

In einer weiteren Spalte namens „Typ“ ist der Verzierungstyp codiert, der auf dem jeweiligen Fund auftritt. Die Codes könnten etwa lauten „A 1“, „A 2“, „A 3“, „B 1“, „B 4“ usw. Der Verzierungstyp ist also auch eine Nominalvariable, deren Ausprägungen wiederum mit Zeichen codiert sind. R hat auch diese Spalte korrekterweise beim Einlesen automatisch als Nominalvariable („Factor“) erfasst.

Die Funddatenbank enthält nun alle Funde, nicht nur die verzierten Scherben, sondern auch die Funde aus Stein. Die Nominalvariable Fundmaterial ist mit ihren beiden Ausprägungen in

der Spalte „Mat“ mit den Codes „ton“ oder „stein“ erfasst. Da die steinernen Funde keine Keramikverzierung aufweisen, ist bei ihnen das Merkmal „Typ“ mit einem „X“ als Kürzel für „Keine Angabe“ codiert. Auch hier wurde korrekterweise eine Nominalvariable („Factor“) eingelesen.

Die CA möchte man auf die verzierten Keramikfunde anwenden. Dafür müssen diese zuerst aus der Datenbank ausgewählt werden. Eine Möglichkeit ist der Befehl ‚subset([Datentabelle], [Variablenname]==“[Ausprägung]“)’. Als erste Argument wird die Datentabelle gesetzt, als zweites die Variable, anhand derer man auswählt. Hier wird nun mit dem doppelten Ist-Gleich-Zeichen ‚==’ die logische Operation ausgedrückt. Bei Nominalvariablen sind nur ‚==’ oder ‚!=’ als Zeichen für Ist-ungleich sinnvoll.

```
subset(data, Mat=="ton")->subdat
```

Betrachtet man nun das Merkmal „Typ“

```
str(subdat$Typ)
```

so sind nach wie vor 10 Ausprägungen („Levels“) vorhanden, obwohl im Auswahlsatz kein Steinfund und folglich auch keine Ausprägung „X“ mehr vorhanden ist. Das Problem wird beseitigt, indem man die Einträge der Spalte „Typ“ mit sich selbst ersetzt, aber dabei den Index ‚[drop=TRUE]’ setzt. Dadurch werden alle nicht vorhandenen Merkmalsausprägungen beim Merkmal „Typ“ weggelassen.

```
(subdat$Typ<-subdat$Typ[drop=TRUE])
```

Jetzt kann man die Kreuztabelle der beiden Nominalvariablen Befund-Nummer („Bef.Nr.“) und Keramikverzierung („Typ“) erzeugen. Mit dem Befehl ‚table()’ lassen sich sowohl die Ausprägungen einer Nominalvariablen auszählen, als auch eine Kreuztabellierung von zwei derartigen Variablen erstellen. So zeigt der folgende Befehl die Häufigkeit der einzelnen Verzierungstypen.

```
table(subdat$Typ)
```

Mit der nächsten Code-Zeile wird eine Kreuztabelle erzeugt und als Objekt abgelegt.

```
table(subdat$Bef.Nr, subdat$Typ)->tab; tab
```

Diese Kreuztabelle entspricht inhaltlich einer Abundanzentabelle mit den absoluten Häufigkeiten der Kombinationen aus den Merkmalen Befundnummer und Verzierungstyp. Einer CA steht jetzt nichts mehr im Wege.

```
ca(tab)->catab
plot(catab, map="rowprincipal")
```

Die Merkmalskombinationen dieses Datensatzes wurden mit reinen Zufallsziehungen erzeugt, weshalb eine Interpretation der CA hier entfallen kann – es ging ja nur um eine Anleitung zum Umgang mit eigenen Daten.

[Stand: Samstag, 29. Januar 2011]