# Gewusst wie: Praxisleitfaden Seriation und Korrespondenzanalyse in der Archäologie

**(2.0)**

Frank Siegmund

2023-01-18

## Table of contents

## Vorwort

Die erste Auflage dieses Leitfadens wurde 2015 u. a. in einer gedruckten Ausgabe publiziert (Siegmund 2015). Seitdem haben sich die Theorie und die methodischen Aspekte der Seriation / Korrespondenzanalyse nicht wesentlich verändert. Insofern ist die Erstauflage weiterhin aktuell. Anlass für die Neuauflage sind einige Anpassungen wie z.B. das Abschneiden einiger “alter Zöpfe” betreffs Software, vor allem aber das Hinzufügen eines zweiten Teils, der das Durchführen einer CA mit Hilfe der Software **R** erläutert (Kap. 19 ff.). Das Umsatteln von PAST auf **R** ist dann sinvoll, wenn die zu bearbeitenden Tabellen unhandlich groß werden, d. h. deutlich mehr Platz einnehmen als 1 bis 2 Bildschirmseiten. Wer dieses Problem nicht hat, kann weiterhin mit der Erstauflage und dem Programm PAST erfolgreich seinen Weg finden.

*Wer mit der Erstausgabe und dem Thema Seriation mit PAST bereits vertraut ist und sich vor allem für die Arbeit mit **R** interessiert, überblättere das Folgende und starte gleich mit Kap. 19.*

## 1 Zielsetzung

Seriation und Korrespondenzanalyse (zu den Begriffen siehe Kap. 5) sind statistische Methoden, die oft auf archäologisches Material angewendet werden, vor allem bei chronologischen Fragestellungen. Der Praxisleitfaden gibt Anfängern eine kurze Einführung in das Verfahren; dazu wird – soweit nötig – die statistische Theorie skizziert und vor allem eine praktische Einführung geboten. Das Buch ist mehr zum Durch- und Nacharbeiten gedacht als zum alleinigen Lesen. Für die hier vorgestellten Übungen und die eigene Praxis braucht es lediglich einen gewöhnlichen Computer, Zugang zum Internet, diesen Leitfaden und etwas Zeit und Energie, um den folgenden Text inklusive der praktischen Übungen gründlich durchzuarbeiten. Nach etwa acht Stunden konzentrierten Selbststudiums ist man kein Anfänger mehr, sondern in der Lage, die einschlägigen Publikationen besser zu verstehen und vor allem auch selbständig eigene Projekte mit “echten” Daten und Fragestellungen durchzuführen.

Für das praktische Üben verwenden wir die kostenlose, gute und ausgereifte Software PAST. Alles technisch Nötige ist hier schnell erlernt, so dass man sich auf das Inhaltliche konzentrieren kann. Wer jedoch anschließend größere Tabellen bearbeiten möchte, wird mit PAST an Grenzen stoßen. Nicht an technische Grenzen der Software, sondern an solche der Bedienbarkeit; denn das Arbeiten mit Tabellen wird, wenn die Menge von ein bis zwei Bildschirmseiten überschritten wird, mühsam und auch fehleranfällig. Daher führe ich im zweiten Teil des Buches (Kap. 19 ff.) – nur soweit nötig – in die freie Software **R** ein, weil damit alternativ zur Datenhaltung als Tabelle eine Dateneingabe und -verwaltung in Form von Listen möglich ist (zum Thema "Listen" s. Kap. 19). Aller für das grundlegende Arbeiten nötige R-Code wird in diesem Buch mitgegeben und erläutert, so dass man – ohne R wirklich lernen zu müssen – nach ca. 6 Stunden weiteren Zeitaufwandes in der Lage ist, mit der für große Tabellen geeigneteren Listenform der Datenverwaltung selbständig zu arbeiten. Wer anschließend mehr zu **R** und Statistik lernen will, sei z. B. auf Siegmund (2020), Kabacoff (2022) oder Field et al. (2012) verwiesen.

## 2 Einleitung

Korrespondenzanalyse – im Folgenden nach der englischen Bezeichnung *correspondence analysis* als CA abgekürzt – ist eine gut begründete multivariate Methode, mit der für die Zeilen und Spalten einer Tabelle eine optimale Reihenfolge gefunden werden kann, wenn die Daten dem unimodalen Modell folgen. Was bedeutet dieser Satz? Der Begriff "multivariat" meint statistische Verfahren, die viele Variablen gleichzeitig berücksichtigen – im Gegensatz zu Verfahren, die nur eine Variable untersuchen (univariat) oder den Zusammenhang zwischen zwei Variablen (bivariat), wie z. B. eine Korrelations- und Regressionsrechnung über den Zusammenhang zwischen Körperhöhe und -gewicht.

Der Begriff "unimodal" bezeichnet Phänomene, bei denen die Werte entlang einer Achse ein Maximum aufweisen und vorher ebenso wie nachher deutlich niedriger ausfallen oder Null betragen. Ein schönes Beispiel für eine unimodale Datenreihe ist die Glockenkurve (**Abb. 1**). Der Begriff unimodales Modell steht im Kontrast zu einem eher linearen Verhalten von Phänomenen ("je mehr von A, desto mehr von B") (**Abb. 2**). Um diesen wichtigen Gegensatz anschaulich zu erklären, wählen wir aus dem Alltag bekannte Beispiele: Die Beziehung zwischen Körperhöhe und Körpergewicht bei Menschen ist tendenziell linear, ebenso die zwischen dem Gewicht eines Autos und seiner Geschwindigkeit einerseits und seinem Benzinverbrauch andererseits. Generell sind größere Menschen auch schwerer, kleinere Menschen auch leichter. Je schwerer ein Fahrzeug ist und je schneller es fährt, desto mehr Benzin verbraucht es. Gewiss handelt es sich bei diesen Beispielen nicht um einfache 1:1 Beziehungen, aber für die nähere Untersuchung solcher Phänomene wählt man lineare Verfahren. Ein gutes Alltags-Beispiel für ein unimodales Phänomen ist die übliche Beziehung zwischen dem Körpergewicht von Menschen und ihrem Lebensalter: Neugeborene sind noch vergleichsweise leicht und werden mit zunehmendem Alter schwerer; viele Menschen weisen ein Gewichtsmaximum irgendwann während ihres Erwachsenenlebens auf und sind im hohen Alter wieder etwas leichter.

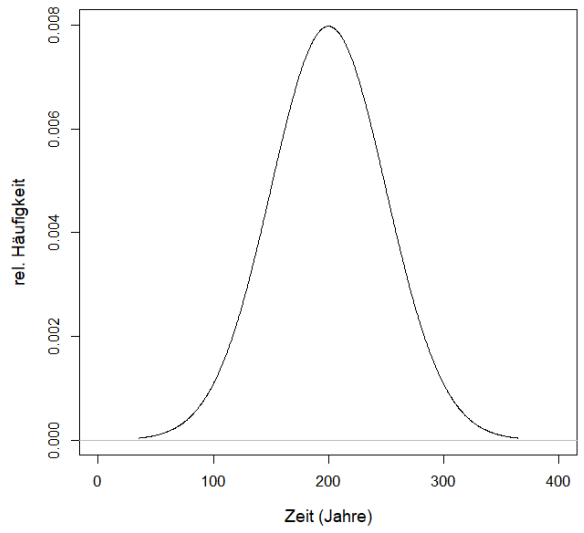
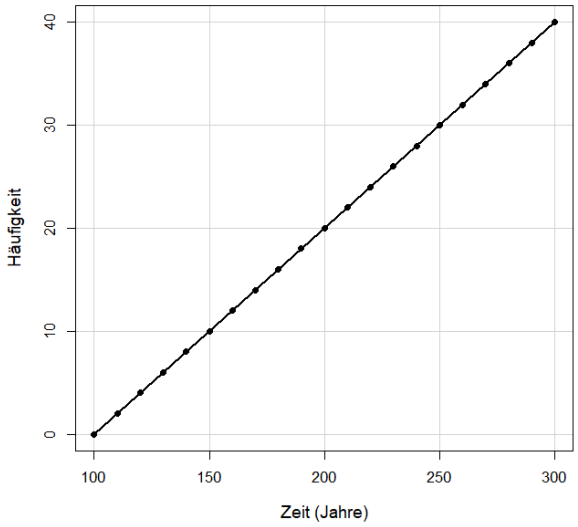



**Abb. 1** Unimodales Modell. Eine glockenförmige Kurve als Ideal für die Häufigkeitsverteilung eines archäologischen Phänomens entlang der Zeitachse. Zu Beginn wird ein neuer Typ gerade erst eingeführt, seine Häufigkeit steigt von Null auf gering. Danach wird er häufiger, ist modern, nach einem Maximum lässt seine Beliebtheit wieder nach bis hin zum völligen Verschwinden.

**Abb. 2** Lineares Modell. Idealbild einer linearen Beziehung: Je mehr die eine Größe wächst, desto mehr wächst auch die andere Größe.

Die noch kurze Geschichte der Speichermedien für Computer ist ein anderes Beispiel für unimodale Modelle, und dieses Beispiel ist bereits sehr nah an den archäologischen Anwendungen der Korrespondenzanalyse. Mechanische Lösungen zur Informationsspeicherung wie etwa Lochstreifen und Lochkarten wurden in den 1960er-Jahren allmählich durch die Speicherung auf großen rotierenden Magnetplatten abgelöst. Nach einigen Jahren der dominanten Verwendung von sog. Winchester-Laufwerken wurden diese in den 1980er-Jahren sukzessive abgelöst durch 8-Zoll Floppy-Disks, dann 5¼-Zoll Floppy-Disks, dann 3½-Zoll Disketten und später durch wiederbeschreibbare CDs bis hin zu den aktuellen USB-Sticks oder mobilen Festplatten (Christensen 1997). Diese von vielen Zeitgenossen zumindest in Teilen erlebte Geschichte ähnelt den Vorstellungen in der Archäologie über Artefakte und Zeit: Ein spezieller Gegenstand – oft als "Typ" bezeichnet – ist noch nicht erfunden, seine Häufigkeit in der Welt beträgt Null. Nach seiner Erfindung und Einführung erscheint er zunächst in geringen Häufigkeiten in der Welt, sobald er sich durchgesetzt hat und "modern" wird, tritt er häufig auf. Später treten neue Objekte auf, die seine Stelle einnehmen, und der von uns beobachtete Typ wird wieder seltener bis hin zu seinem völligen Verschwinden, d. h. seine Häufigkeit geht auf Null zurück (wie **Abb. 1**). Für die Analyse derartiger Phänomene ist die CA das Verfahren der



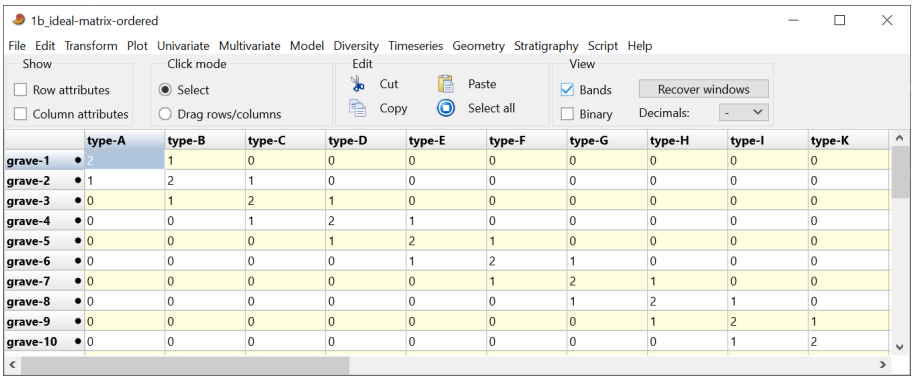

Wahl. Dabei ist es nicht erforderlich, dass wie bei einer Glockenkurve – Statistiker sprechen hier von einer "Normalverteilung" – eine exakte Symmetrie des Bildes gegeben ist. Der Begriff unimodal erwartet nur, dass es ein Maximum irgendwo innerhalb der untersuchten Reihe gibt, während an beiden Enden Minima beobachtet werden. Abweichungen vom Idealbild einer Glockenkurve sind erlaubt und beeinflussen das Ergebnis einer CA nicht schwerwiegend.

Erscheinungen, die dem unimodalen Modell folgen, sind weder exotisch noch auf die Archäologie oder das Phänomen Zeit beschränkt. Andere Beispiele für unimodale Modelle bieten z. B. Pflanzen und Tiere, die für ihr Leben bestimmte Umweltbedingungen bevorzugen, d. h. eine bestimmte Temperatur, Feuchtigkeit, Lichtexposition, Bodensäure etc. Unter den für sie optimalen Bedingungen sind sie in der Natur häufig; sie werden seltener, wenn dieses Optimum verlassen wird, und zwar unabhängig davon, in welche Richtung vom Optimum abgewichen wird (z. B. sowohl deutlich kälter/nasser als auch deutlich wärmer/trockener). Entlang der Umweltbedingungen zeigt die Häufigkeit vieler Lebewesen ein unimodales Verhalten.

Eine andere willkommene Eigenschaft der CA ist ihre Robustheit. Sie ist voraussetzungsarm und kann auf viele Arten von Daten angewendet werden. Die CA kann mit Anwesenheits- und Abwesenheitsinformationen umgehen, Häufigkeiten (statistisch "Nominalskala") oder Ränge ("Rangskala") analysieren, aber auch auf Messwerte angewendet werden, während viele andere multivariate Verfahren Messwerte ("Intervallskala", "Verhältnisskala") erfordern. Dies ist ein weiterer Grund für die Beliebtheit der CA in der Archäologie, da hier oft Anwesenheits-/Abwesenheits-Beobachtungen oder Häufigkeiten vorliegen. Indes: die CA ist nicht auf die Archäologie begrenzt. Vielmehr ist sie auch in vielen anderen Wissenschaften eine beliebte statistische Methode, wie etwa in den Sozialwissenschaften, der Biologie oder Ökologie. Das bekannte Werk des französischen Soziologen Pierre Bourdieu "Die feinen Unterschiede. Kritik

der gesellschaftlichen Urteilskraft" (1979, deutsch 1982) ist ein interessantes Beispiel für eine frühe Anwendung der CA in den Sozialwissenschaften.

Plant man die Anwendung multivariater Verfahren zur Analyse von Daten und ist unsicher, ob die Daten eher dem unimodalen Modell (**Abb. 1**) oder einem linearen Modell (**Abb. 2**) folgen, ist es nützlich, die methodischen Alternativen für lineare Modelle zu kennen. Angemessen und vergleichbar zur CA sind für Daten, die dem linearen Modell folgen, die Hauptkomponentenanalyse (PCA, *principle component analysis*) oder die Faktorenanalyse (FA, *factor analysis*); sie sind – ähnlich der CA – ordnende Verfahren, im Unterschied z. B. zu Clusteranalysen als gruppierende Verfahren. Die Anwendung einer Hauptkomponentenanalyse (PCA) auf unimodale Daten führt zu irrigen Ergebnissen, ebenso wie die Anwendung einer CA auf lineare Daten. In Kap. 12.5 werden wir uns dies an einem Beispiel anschauen. In der Praxis erweisen sich PCA und FA als empfindlicher gegen leichte Verletzungen ihrer Anforderungen an die Daten, während die CA robuster gegenüber solchen Abweichungen vom idealen Modell ist.




| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-3 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 |
| grave-8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |

**Abb. 3** Beispiel einer ideal diagonalisierten Tabelle (Matrix), in der die Zeilen und die Spalten dem unimodalen Modell folgen.

## 3 Theorie und Zielsetzung der Korrespondenzanalyse

Wirft man einen Blick auf den aktuellen (2023) Artikel *correspondence analysis* in der englischsprachigen Wikipedia, erscheint die CA als etwas höchst Kompliziertes. Dieser Eindruck täuscht. Die CA ist einfach zu verstehen, zu rechnen und durchzuführen. Gleichwie, eine gründliche Einführung in die Theorie und die Berechnung ist hier nicht notwendig, denn es gibt gute weiterführende Bücher, die dies bereits leisten (Kap. 5). Das eigenhändige Rechnen oder Programmieren einer CA ist heutzutage ebensowenig nötig, weil es Computer und Software gibt, welche diese Aufgabe übernehmen (Kap. 4). Wir wollen uns daher zunächst darauf konzentrieren, die Zielsetzung der CA zu verstehen: Es geht darum, die Zeilen und Spalten einer gegebenen Tabelle so neu zu ordnen, dass die Zahlen in dieser Tabelle am Ende eine Diagonale bilden. Üblicherweise bestehen solche Tabellen (auch: Kontingenztafel, engl. *contingency table*) aus einer Vielzahl von leeren Zellen oder Nullen und wenigen anderen Zellen

mit Einsen oder Häufigkeiten. Nach der Neuordnung der Tabelle sollten all diese Einsen resp. Häufigkeiten sich entlang einer Diagonalen in der Mitte der Tabelle häufen (**Abb. 3**). Mit Erreichen dieser neuen Ordnung folgen die Werte in den Zellen – jeweils zeilen- oder spaltenweise gelesen – dem unimodalen Modell: Jede Zeile und jede Spalte zeigt zunächst Nullen resp. leere Zellen, dann diverse Einsen oder Häufigkeiten, und anschließend wieder Nullen oder leere Zellen. Eine CA beginnt also mit einer ungeordneten Tabelle und ihr Ergebnis ist eine nach dem unimodalen Modell optimal neu geordnete Tabelle.

Unmittelbar nach Erfindung des Verfahrens geschah dieses Sortieren tatsächlich durch ein sukzessives mechanisches Umordnen der Zeilen und Spalten einer Tabelle mit der Hand. Zunächst wird die Ordnung der Zeilen optimiert, dann die der Spalten, dann wiederum die der Zeilen usw., bis sich eine stabile Lösung mit einer guten Diagonalen ergibt und das wiederholte Umordnen abgebrochen werden kann. Weil dabei die Position der Spalten oder Zeilen jedesmal wieder neu gemittelt wird, spricht man auch vom *reciprocal averaging*. Die ganze Methode wurde oft als Seriation (engl. *seriation*, *ordination*) oder als *sequencing* resp. *sequence dating* bezeichnet. Fotos einer konsequent ausgearbeiteten mechanischen Lösung dieser Aufgabe finden sich bei Périn (1980, Abb. 23). Die erste computergestützte Lösung für eine CA war nur eine Automatisierung dieses mechanischen Prozesses, d. h. per Computer wurde nach einem sehr einfachen Rechenverfahren (Ihm 1983) eine wiederholte Umsortierung aller Zeilen und Spalten einer Tabelle vorgenommen (Goldmann 1972). Heutzutage ist das Rechenverfahren besser ausgearbeitet, mathematisch feinsinniger und beruht allein auf Berechnungen, d. h. erst am Ende des Verfahrens wird die Tabelle aufgrund der Ergebnisse einmal neu geordnet.

## 4 Software zur Durchführung einer Korrespondenzanalyse

Bei vielen der heute für eine CA verwendeten Computer-Programme handelt es sich um freie oder sogar quell-offene Software (Open Source), sodass dem Anwender keine besonderen Kosten entstehen. Man muss also nur lernen, mit diesen Programmen umzugehen. Die nachfolgend vorgestellte Liste ist eine persönliche Auswahl des Autors, der selbst sehr gründliche und langjährige Erfahrungen mit WinBASP und PAST besitzt. Alle genannten Programme werden mit guten Anleitungen verteilt, anhand derer ihre Bedienung leicht erlernbar ist. Ich empfehle sehr, die jeweiligen Handbücher eingehend zu studieren. Dieser Praxisleitfaden verwendet PAST für die praktischen Übungen, es sei jedoch betont, dass andere Softwarelösungen ebenfalls gut sind und jeweils spezifische Vorzüge haben. Näheres zur Software-Auswahl bietet Kap. 6. Neben aktuellen Programmen liste ich auch Software, die z. B. in den 1980er- und 1990er-Jahren viel verwendet wurde, heute aber nicht mehr zeitgemäß ist.

Dieser Praxisleitfaden fokussiert auf Computer mit dem Betriebssystem MS-Windows. Einige der hier angeführten Programme wie z. B. CAPCA oder WinBASP stehen nur für MS-Windows zur Verfügung. Das hier verwendete PAST ist jedoch auch auf MAC-Computern lauffähig, ebenso das mächtige Statistik-Werkzeug “R”, das zudem auf Linux-Systemen arbeitet.

Caveat: Dieser Text wurde im Jahr 2023 verfasst. Alle hier erwähnte Software resp. die Links dorthin wurden zuletzt im Januar 2023 benutzt und überprüft. Sollten sich danach insbes. die Links geändert haben, müssten es die verwendeten Begriffe ermöglichen, das Gesuchte mit den einschlägigen Suchmaschinen selbst schnell zu finden.

Alle wichtigen im folgenden angeführten Programme für die Durchführung einer CA werden in einer von Martin Hinz (Univ. Bern) 2012 aufgelegten Reihe von Videos vorgestellt. Für Interessierte ist dies ein ausnehmend nützlicher Fundus, durch den man mit geringem Zeitaufwand gründliche Eindrücke von den Programmen gewinnen kann, ohne diese installieren oder ggf. gar kaufen zu müssen. Weiteres dort: https://vitutr.archaeological.science/tags/correspondence_analysis/ [6.1.2023].

### PAST 4.1: Paleontological Statistics Version 4.1 (von Øyvind Hammer)

Quelle: https://www.nhm.uio.no/english/research/resources/past/ [6.1.2023].

PAST 4.1 wird vor allem für Computer unter MS-Windows (11, 10, 8, 7) entwickelt, es gibt aber auch eine Version für Mac-Computer (OSX Catalina). Die aktuelle Version PAST 4.1 ist sehr ausgereift und stabil, wird aber vom Autor immer wieder aktualisiert. Zu jeder neuen Version von PAST wird auch das elektronisch verfügbare Handbuch aktualisiert; man findet es auf der o.g. Website von PAST. Um PAST wissenschaftlich zu zitieren, nenne man: Hammer, Harper & Ryan (2001).

### R-Pakete zur Korrespondenzanalyse

Eine Korrespondenzanalyse ist im Basisumfang des mächtigen und umfangreichen Open-Source-Statistikpakets **R** nicht enhalten. Aber **R** kann durch sog. "Pakete" erweitert werden und es gibt mehrere gute und stabile R-Pakete zur Durchführung einer CA. Alle mir bekannten Pakete für eine CA mit **R** finden sich in dessen offiziellem Paket-Archiv CRAN ("The Comprehensive R Archive Network"; dort: https://www.r-project.org/). Zu jedem Paket auf CRAN gehört eine eigene Dokumentation, die man ggf. konsultieren sollte.

Eine gute Anleitung zur Durchführung eine Korrespondenzanalyse mit dem R-Paket "CA" findet sich auf der Website "Rchaeology" von Georg Roth (FU Berlin): http://www.rchaeology.eu/ Ein R-Skript zur Durchführung einer CA zusammen mit einer guten Einführung in englischer Sprache findet sich bei: http://cainarchaeology.weebly.com/ (Alberti 2013; 2015).

Eine beliebte grafische Nutzeroberfläche ("GUI") für **R** ist der R-Commander, für den es wiederum ein Plugin gibt, mit dem u. a. eine CA gerechnet werden kann. Weiteres zu diesem Plugin "FactoMineR" dort: http://factominer.free.fr/graphs/RcmdrPlugin.html [6.1.2023].

Diese Hinweise genügen. Weiteres zum Thema "Durchführung einer CA mit **R**" im zweiten Teil des Buches (Kap. 19 ff.), wo das Arbeiten mit dem R-Paket CA vertieft wurd.

### WinSERION 3.1 (von Peter Stadler)

WinSerion ist Teil der von Peter Stadler (Wien) seit den 1980er-Jahren entwickelten und weiterhin gepflegten "Serion Suite". In seiner Funktionalität betreffs CA ähnelt WinSerion sehr dem Programm WinBASP, bietet aber mit zahlreichen Zusatzelementen weit über die reine CA hinausgehende Funktionen. Der kostenlose Zugang zu WinSerion ist auf Teilnehmer an den Lehrveranstaltungen von P. Stadler limitiert (u.a. Univ. Wien, München), weiteres entnehme man der Website: https://www.winserion.org/index.html [6.1.2023].

### CANOCO 5 (von Cajo J. F. ter Braak, Univ. Wageningen)

Canoco ist ein seit den 1980er-Jahren entwickeltes spezielles Softwarepaket für CA inkl. vieler Varianten der CA, das aber bezahlt werden muss (Anfang 2023: ca. 400 €). Die Version 5 wurde 2012 herausgegeben, die aktuelle Version 5.12 im Okt. 2019. Weitere Informationen dort: https://de.wikipedia.org/wiki/Canoco und dort: https://www.wur.nl/en/research-results/research-institutes/show/canoco-for-visualization-of-multivariate-data.htm Nach der Emeritierung von ter Braak wird das Programm nicht mehr von der Univ. Wageningen unterstützt, sondern von dessen Programmierer Petr Šmilauer gepflegt und vertrieben; näheres dort: https://www.canoco5.com (Achtung, Seite reagiert bisweilen langsam).

In den 1980/90er-Jahren bot CANOCO einige Funktionen, die anderweitig nicht / kaum / nur schwierig verfügbar waren, z.B. eine kanonische Korrespondenzanalyse (z.B. Siegmund 1991; 1994). Dieser methodische Vorsprung besteht heute im Vergleich z.B. zu PAST oder einschlägigen R-Paketen nicht mehr.

### Wichtige nicht mehr aktualisierte Programme

### WinBASP: The Bonn Archaeological Software Package, Version 5.43 (von Irwin Scollar u.a.)

Quelle, ehedem: http://www.uni-koeln.de/\~al001/

heute z. B.: https://web.archive.org/web/20150216062559/http://www.uni-koeln.de//~al001/

Der Name WinBASP steht für ein seit 1973 entwickeltes und seinerzeit weit verbreitet benutztes Programmpaket, dessen weitere Entwicklung in etwa mit dem Aufkommen von Windows 7 eingestellt wurde. WinBASP lief zuletzt gut und stabil unter dem Betriebssystem MS-Windows XP, aber nicht mehr unter 64-Bit-Betriebssystemen wie MS-Windows 7 ff.

Für das Arbeiten mit wirklich umfangreichen Datensätzen habe ich WinBASP ungemein geschätzt. Das Einüben in das Programm verläuft dank des mitgelieferten Handbuchs schnell, nach kurzem Üben kann man es routiniert bedienen. Folglich schrieb ich 2015 in die Erstauflage dieses Leitfadens: "Müßte ich ein großes Seriations-Projekt starten und hätte ein Problem mit

WinBASP wegen eines 64-Bit-Betriebssystems auf meinem normalen Arbeitsplatz-Computer, würde ich – nur für die CA – ernsthaft das Arbeiten mit einem alten Gebraucht-PC erwägen: Ihn dauerhaft vom Netz abhängen, ihn nach Möglichkeit mit dem Betriebssystem Windows XP Professional (SP 3) versehen, WinBASP aufspielen, und dort meine Seriationen und Korrespondenzanalysen rechnen. Den geringen Hardware-Kosten eines solchen Seriations-PCs steht ein großer Gewinn an Arbeits-Effizienz gegenüber." Acht Jahre und einige MS-Windows-Generationen später dürfte dies kein praktikabler Weg mehr sein, weshalb ich nun für solche Anwendungsfälle empfehle, den Weg mit **R** und R-Paketen zu gehen - so, wie ich es im zweiten Teil dieses Leitadens beschreiben werde.

Hinweis für ehemalige WinBASP-Nutzer: Auf der o.g. Website zu WinBASP findet sich ebenfalls ein Programm "BaspPast", das einen einfachen Datenaustausch zwischen WinBASP und PAST erlaubt – und damit auch zwischen WinBASP und anderen Tabellenkalkulationsprogrammen wie z. B. LibreOffice Calc oder MS-Excel. Indes: nach den Erfahrungen des Autors hat BaspPast einen kleinen Programmierfehler beim Datenexport von WinBASP nach PAST, denn der letzte Typ einer Liste verschwindet beim Export nach PAST. Kein Problem, wenn man diesen Fehler kennt, denn es gibt eine einfache Lösung: Vor dem Datenexport nach PAST wird in WinBASP noch ein letzter, neuer (fiktiver) Typ angelegt. Nach dem Export ist er verschwunden, aber alle anderen Daten sind so übertragen, wie es geplant war. ;-)

### CAPCA 3.1 (von Torsten Madsen)

Quelle: https://www.archaeoinfo.dk/ [6.1.2023].

Insbesondere viele skandinavische und britischen Archäologinnen arbeiteten mit dieser bewährten Lösung, die als Add-in in das Programm MS-Excel eingebunden wird, d. h. ein installiertes MS-Office voraussetzt. Mit CAPCA können eine CA und eine PCA berechnet werden. Die jüngste Version CAPCA 3.1 (vom 11.2.2016) arbeitet mit MS-Excel 2007 (ff.), eine ältere Version CAPCA 2.2 für MS-Excel 2003 ist weiterhin verfügbar. Ein zusammen mit CAPCA verbreitetes, ca. 30-seitiges englischsprachiges Handbuch führt in die Installation und Bedienung ein (vgl. auch Madsen 2007). Das AddIn wird nach Aussage des Autors nicht mehr aktualisiert, d.h. eine 64-Bit-Version wird es nicht geben.

## 5 "Seriation" oder "Korrespondenzanalyse": Name der Methode und einführende Literatur

Die Methode der Seriation / Korrespondenzanalyse wurde zu unterschiedlichen Zeiten mehrfach erfunden und folglich auch unterschiedlich bezeichnet (dazu Ihm 2005; de Leeuw 2013). In Deutschland beispielsweise wurde das Verfahren als Seriation bezeichnet, als es von Klaus Goldmann und Ernst Kammerer in die Archäologie eingeführt wurde (Goldmann 1972). Sie entlehnten den Begriff von Sir Flinders Petrie (1853-1942), der im späten 19. Jahrhundert ein ähnliches Vorgehen als *seriation* bezeichnet hatte (Petrie 1899), wobei sein Vorgehen mehr intuitiv ohne Mathematik und strengen Algorithmus erfolgte. Heute benutzt

die Fachwelt wie üblich jene Bezeichnung, die der Autor vorschlug, der als Erster grundlegend das moderne Verfahren entwickelte, nämlich der französische Statistiker Jean-Paul Benzécri (1976) – also Korrespondenzanalyse.

Die heute verfügbare Literatur zu diesem Verfahren ist in ihrer Fülle überwältigend. Will man zur Vertiefung dieses Leitfadens so wenig wie möglich und so viel wie nötig lesen, empfehle ich das Buch von Michael J. Greenacre (2007), das auch viele praktische Hinweise enthält, während Greenacre (1984) meist als das grundlegende Standardwerk zitiert wird, wenn es um eine theoretische Einführung in die CA geht.

## 6 Beginn des praktischen Teils: die Wahl der Software

Eine Korrespondenzanalyse durchführen heißt, mit Tabellen zu arbeiten, welche Gräber oder andere geschlossene Funde wie z. B. Siedlungsgruben darstellen, die Typen enthalten. Eine CA durchführen meint nicht: Die Daten in eine Tabelle eingeben, einmal rechnen, danach neu sortieren und fertig. Vielmehr bedeutet es, die resultierenden Tabellen immer wieder neu auch archäologisch zu analysieren, dabei nach Mängeln und Optimierungsmöglichkeiten zu fahnden, dann den Datenpool zu verändern, neu zu rechnen und zu schauen, ob und was sich, gemessen an der Zielsetzung, verbessert oder verschlechtert hat. Bei kleinen Tabellen, wie wir sie hier zum Üben benutzen, ist das kein Problem. Bei großen Tabellen, d. h. etwa ab 30 Spalten und 50 Zeilen (i.e. ca. 1 Bildschirmseite), wird dies zunehmend mühsam und unübersichtlich, letztlich auch fehleranfällig. Daher schätzte ich das Programm WinBASP sehr, auch wenn es aus heutiger Sicht altbacken ausschaut. Denn WinBASP umfasst mächtige Werkzeuge zur Dateneingabe und -verwaltung, die das Datenmanagement sehr erleichtern und fehlerarm werden lassen. Vor allem beruht die Eingabe der Informationen auf Listen anstelle von Tabellen, was der üblichen Organisation im Schrifttum oder bei der Primäranalyse von Material und dem Erarbeiten einer Typologie ähnelt (Typ xyz kommt vor in den Befunden a, b, c, …). Mit WinSERION habe ich persönlich keine praktischen Erfahrungen, aber sein Datenmanagement folgt einem ähnlichen Konzept. Wer immer ein umfangreiches Projekt mit vielen Gräbern oder Befunden und Typen ins Auge fasst, sollte sich Software, die eine Listeneingabe der Daten erlaubt, anschauen und ihren Einsatz erwägen.

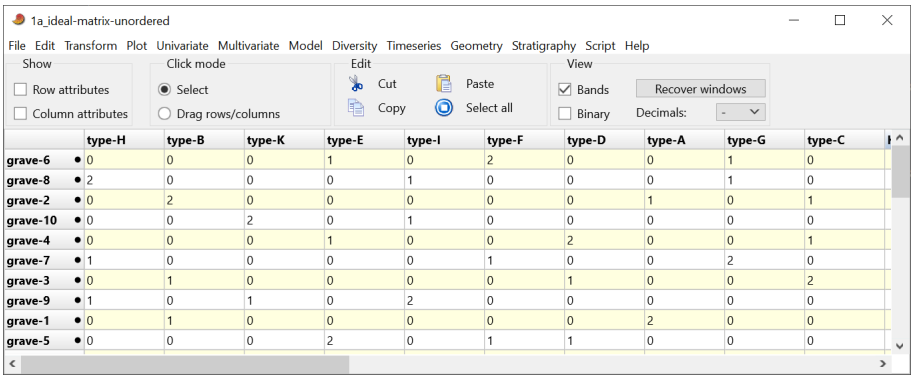
Wenn nur eine kleine Tabelle zu analysieren ist, empfehle ich PAST. Es ist modern, gut gemacht, kostenlos, und es bietet jenseits der CA auch umfangreiche weitere statistische Prozeduren an. Man braucht lange, bis man an die Grenzen von PAST stößt. Wer eines der gängigen Tabellenkalkulationsprogramme beherrscht wie z. B. LibreOffice Calc oder MS-Excel, dort die bei PAST nur in engen Grenzen mögliche Datenverwaltung löst und für die Statistik dann PAST benutzt, kann viele archäologische Probleme lösen, ohne zu komplexeren Werkzeugen greifen zu müssen. Naheliegenderweise gründet diese Einführung daher auf PAST. Wer dem im nächsten Kapitel beginnenden praktischen Teil folgen möchte, lade daher PAST von der genannten Website herunter und installiere es auf seinem Computer.

Seien Sie nicht verwirrt: PAST hat keine aufwändige Installationsprozedur, so wie man es von vielen anderen Windows-Programmen kennt, sondern PAST ist nach dem Entpacken einfach eine einzige fertige *.exe-Datei. Nach dem Klicken auf das Symbol startet PAST, das war's. Kleiner Tipp: die Arbeit mit PAST wird unkomplizierter, wenn man das Programm auf seinem Computer im gleichen Ordner ablegt, in dem man auch die Daten hält, und ggf. auch einfach zwischen verschiedenen Ordnern mit unterschiedlichen Datensammlungen/Projekten hin- und herschiebt.

Der praktische Teil dieses Leitfadens beruht darüber hinaus auf ein paar einfachen Übungsdaten. Man kann diese Daten aus den Abbildungen dieser Broschüre direkt in PAST eintippen, man kann sie aber auch von der Website des Autors herunterladen (http://www.frank-siegmund.de, » Veröffentlichungen, » Open Data), oder von dessen Archiv bei Academia.edu.

## 7 Der Start mit PAST

Man starte PAST mit einem Doppelklick auf das Programmsymbol. Man gehe zu File, dann zu *Open*, und öffne den Datensatz 1a_ideal-matrix-unordered (**Abb. 4**) – oder gebe die Daten entsprechend **Abb. 4** direkt in die Tabelle von PAST ein. Sie sollten nun ein von Tabellenkalkulationsprogrammen wie LibreOffice Calc oder MS-Excel weitgehend vertrautes Bild sehen. Die Beispieltabelle enthält zehn Typen (type-A bis type-K) und zehn Gräber (grave-1 bis grave-10). Wie üblich (aber für eine CA nicht zwingend notwendig) sind hier die Befunde/Gräber als Zeilen angelegt und die Typen als Spalten. Im Prinzip enthält jedes Grab unserer Übungstabelle drei Typen, und jeder Typ ist in drei Gräbern vertreten. Dabei wird in unserem Beispiel mit Häufigkeiten gearbeitet; neben vielen Zellen mit einer Null signalisieren die Ziffern also, wie oft dieser Typ in dem jeweiligen Grab vorkommt.




| | type-H | type-B | type-K | type-E | type-I | type-F | type-D | type-A | type-G | type-C |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-6 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 0 |
| grave-8 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| grave-2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| grave-10 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 1 |
| grave-7 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 |
| grave-3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| grave-9 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| grave-1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 |

**Abb. 4** Bildschirmfoto unseres ersten praktischen Beispiels: die nach PAST eingelesene Eingabetabelle.

Die Tabelle ist vielleicht etwas leichter lesbar, wenn man bei PAST rechts oben unter View das Kästchen *Bands* anklickt: sie wird jetzt mit abwechselnd weißen und gelblichen Zeilen

dargestellt. Diese Einstellung hat keinerlei inhaltlich-statistische Bedeutung, es geht allein um die Lesbarkeit.

### Ein paar grundlegende Bedienungshinweise zu PAST

Wie schon erwähnt: zu PAST gibt es ein gutes und wirklich nützliches Handbuch. Aber damit wir uns hier auf die CA fokussieren können, schnell ein paar grundlegende Hinweise zu seiner Bedienung. Für die Bedienung wichtig ist die oberste Zeile mit *File, Edit* usw., hinter denen sich jeweils nach dem Draufklicken ein aufklappendes und weitgehend selbsterklärendes Menü verbirgt. Der zweizeilige Block darunter zeigt die Bereiche *Show, Click mode, Edit* und *View*. Dabei ist wichtig zu wissen, dass man in PAST entweder editiert, d. h. Daten eingibt bzw. verändert, oder analysiert. Analysiert wird stets nur das, was markiert wurde und in der Datentabelle entsprechend hellblau unterlegt ist. Ist bei *Click mode* der Knopf *Select* aktiviert (blau), kann man die zu analysierenden Bereiche markieren: einfach die Spalte anklicken, das war's. Zur Kontrolle kann man ganz oben auch *Univariate*, dann *Summary statistics* anklicken, dann erhält man univariate Statistiken zu der aktivierten, blau markierten Spalte. Man benötigt die Statistiken nicht für die ganze Spalte, sondern nur für die ersten Fälle? Kein Problem: Mit dem Pfeil/Cursor in das erste gewünschte Feld klicken, dann auf der Tastatur die Shift-Taste (die ansonsten für das Umschalten auf große Buchstaben genutzt wird) drücken und gedrückt halten, mit dem Pfeil / Cursor in die unterste gewünschte Zelle klicken, das war's - der gewünschte Bereich sollte jetzt markiert sein. Überprüfen: *Univariate Statistics*, » *Summary statistics*: jetzt sollten die Statistiken nur für die blau markierten Zellen ausgespielt werden. Genau so lassen sich auch mehrere Spalten gleichzeitig auswählen oder ein Teilfeld innerhalb der Tabelle. Will man mit einem Klick die gesamte Tabelle markieren – so, wie wir es für unsere CA benötigen – klickt man der Einfachheit halber oben im Block *Edit* auf den Knopf *Select all*: die ganze Tabelle wird hellblau unterlegt, sie ist im Ganzen ausgewählt.

Nun studieren wir noch den Bereich *Show* mit den Schaltern *Row attributes* und *Column attributes*. Ein Klick auf *Row attributes* öffnet drei neue Spalten; hier relevant ist lediglich die Spalte *Name*, wo der Name der Spalte eingetragen ist, in unserem Fall also die Grabnummer. Ohne Eintrag dort wird von PAST einfach die Zeilenummer angezeigt. Nehmen wir das Häkchen von den *Row attributes* weg und setzen es bei *Column attributes*, erscheint eine ähnliches Bild für die Spaltenköpfe; nun könnte man bei *Name* die Bezeichnung eines Typs verändern oder neu eingeben. Das war's für unsere Zwecke. Wer PAST tiefer kennenlernen will, sollte das Handbuch studieren oder sich per Versuch und Irrtum einarbeiten.

### Über leere Zellen und Nullen

In der Statistik ist es normalerweise sehr wichtig, zwischen (beobachtet) Null und leerer Zelle (d.h. keine Beobachtung) zu unterscheiden. Drei Geldbörsen mit beobachtet Null, 10 und 20 Euro Inhalt ergeben einen Mittelwert von 10 Euro. Drei Geldbörsen mit einmal unbeobachtet, dann 10 und 20 Euro Inhalt ergeben einen Mittelwert von 15 Euro, bei einem unbeobachteten

Fall. Bei einer CA ist dies anders, denn hier gibt es keinen Unterschied zwischen beobachtet Null und leerer Zelle. Wer zweifelt, möge dies z. B. anhand der Beispiele bei Ihm (1983) eigenhändig nachrechnen. PAST erwartet für die CA in allen Zellen Werte, daher sind in unseren Übungstabellen alle leeren Zellen mit Nullen gefüllt.

## 7.1 PAST Schritt 1: CA rechnen und Streuungsdiagramm lesen

Genug der Vorreden, rechnen wir jetzt schnell eine CA! Dazu wird der zu aktivierende Bereich markiert (siehe oben), was wir der Einfachheit halber mit Hilfe des Schalters *Edit* und *Select all* tun können, oder indem wir mit dem Cursor auf das linke obere Feld klicken, die Shift-Taste drücken (und halten) und einem Klick in das rechte untere Feld abschließen. Die Tabelle sollte jetzt komplett hellblau hinterlegt sein. Nun in die oberste Zeile gehen zu *Multivariate*, » *Ordination*, » *Correspondence (CA)* und klicken (**Abb. 5**). Ein zweites Fenster Correpondence analysis öffnet sich auf dem Bildschirm (**Abb. 6**).

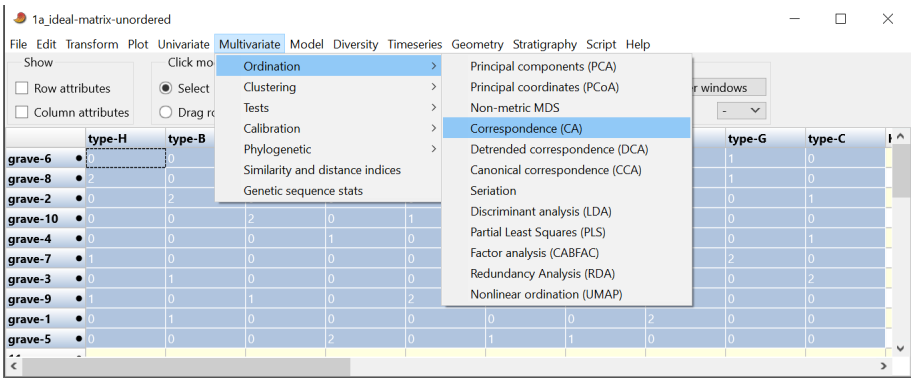
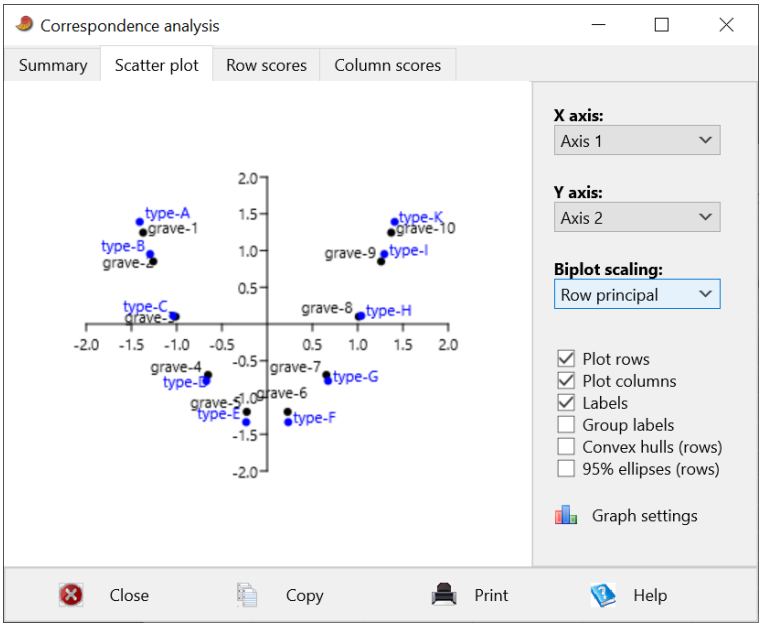



**Abb. 5** Bildschirmfoto von PAST unmittelbar vor dem Berechnen der Korrespondenzanalyse.

Das war's, die CA ist bereits komplett gerechnet. Schauen wir uns die Ergebnisse an. Das neue Fenster zeigt die Reiter *Summary*, *Scatter plot*, *Row scores* und *Column scores*. Per Voreinstellung sollte die Schaltfläche *Scatter Plot* aktiv sein, wenn nicht: draufklicken (**Abb. 6**). Man kann das ganze Fenster durch Klicken am Rand samt Aufziehen vergrößern und damit besser lesbar machen. Das Streuungsdiagramm zeigt die Gräber (schwarz) und Typen (blau) im Raum der beiden Ordnungen Achse 1 (waagerecht) und Achse 2 (senkrecht) an, die durch die Korrespondenzanalyse errechnet wurden. Insgesamt sollten die Punkte in etwa die Form einer Parabel oder eines Hufeisens bilden. Rechts in diesem Fenster kann man die Voreinstellungen verändern – wenn gewünscht. Wenn man z. B. das Häkchen vor *Plot columns* wegnimmt, werden die Spalten (Typen) ausgeblendet und man kann die Gräber besser erkennen.

**Abb. 6** Bildschirmfoto des neuen, zweiten (Ausgabe-)Fensters von PAST. Es zeigt das Streuungsdiagramm von Achse 1 (waagerecht) gegen Achse 2 (senkrecht).



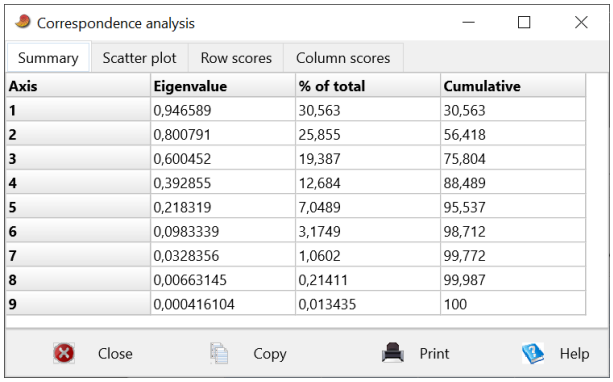

In unserem Beispiel sprechen wir von Gräbern und Typen. Aber eine CA ist nicht auf die Analyse von Grabfunden beschränkt. Das Verfahren wurde ebenfalls erfolgreich auf Horte angewendet und auf Siedlungsbefunde und -schichten und deren Fundinhalte. Ein anderer Anwendungsfalls sind Objekte (als geschlossene Funde) und deren Merkmale, die geordnet werden sollen.

## 7.2 PAST Schritt 2: Tabelle neu ordnen und analysieren

Das Streuungsdiagramm zeigt als Achse 1 die waagerechte x-Achse und als Achse 2 die senkrechte y-Achse (**Abb. 6**). Die x-Achse ist die erste, dominante Lösung der CA. Die zweite Achse gibt eine weitere, weniger bedeutende Ordnung wieder, die unabhängig von der ersten Achse ist. Wir fokussieren wie üblich erst einmal auf die erste Ordnung und lesen die Achse von links nach rechts: Grab-1, Grab-2, Grab-3, … bis Grab-10, eine Reihenfolge, die von der in unserer Eingabetabelle 1a_ideal-matrix-unordered deutlich differiert. Schauen wir auf die Typen: wir nehmen – rechts im Fenster – mit einem Klick auf *Plot columns* das Ausblenden der Spalten (Typen) weg und blenden mit einem Klick auf *Plot rows* das Häkchen dort aus und damit die Zeilen (Gräber). Erneut lesen wir entlang der x-Achse von links nach rechts: Typ-A, Typ-B, Typ-C, … bis Typ-K. Dies ist die Ordnung der Typen nach der ersten Achse der CA, die aufgrund unserer Ausgangstabelle berechnet wurde.

Blicken wir zunächst auf die übrigen Reiter im Fenster *Correspondence Analysis* (**Abb. 7**). *Row scores* (anklicken) zeigt für jedes Grab den Wert entlang Achse 1, 2, 3 usw., *Column scores* entsprechend für jeden Typ seinen Wert auf Achse 1, 2, 3 usw. Diese Scores (früher in der deutschen Literatur auch “Schwerpunkte” genannt) sind die statistischen Werte, die

im Streuungsdiagramm Achse 1 gegen Achse 2 angezeigt worden waren, resp. nach denen die Punkte in das Diagramm gesetzt wurden. Auf die nähere Bedeutung dieser Achsen kommen wir später noch zurück, denn hierzu braucht es eine genauere Erklärung (siehe Kap. 8). Der Reiter oben ganz links *Summary* zeigt die Achsen 1, 2, 3 usw. und deren *Eigenvalue* (Eigenwert), *% of total* (Anteil am Gesamten) und *Cumulativ*, den von Achse zu Achse aufaddierten Anteil am Gesamten. Ja, auch diese Begriffe werden in Kürze erläutert (Kap. 8).



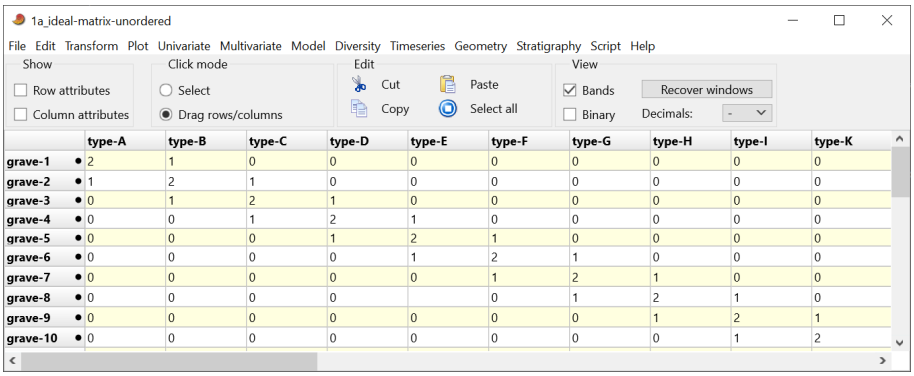

| Axis | Eigenvalue | % of total | Cumulative |
|---|---|---|---|
| 1 | 0,946589 | 30,563 | 30,563 |
| 2 | 0,800791 | 25,855 | 56,418 |
| 3 | 0,600452 | 19,387 | 75,804 |
| 4 | 0,392855 | 12,684 | 88,489 |
| 5 | 0,218319 | 7,0489 | 95,537 |
| 6 | 0,0983339 | 3,1749 | 98,712 |
| 7 | 0,0328356 | 1,0602 | 99,772 |
| 8 | 0,00663145 | 0,21411 | 99,987 |
| 9 | 0,000416104 | 0,013435 | 100 |

**Abb. 7** Das PAST-Fenster Abb. 6, aber mit aktiviertem Reiter "Summary", die numerischen Resultate der CA anzeigend. Für eine nähere Erklärung der Werte siehe Kap. 8.1.

Doch erst einmal versuchen wir, die Ausgangstabelle nach den Ergebnissen der CA neu zu ordnen. Also blicken wir auf das Fenster *Correspondence analysis* mit dem Reiter *Scatter plot* (ggf. anklicken), um die Grafik erneut anzuzeigen. Beginnen wir mit den Typen, die dort in blauer Schrift angezeigt sind. Kehren wir zum Ausgangsfenster mit der Tabelle zurück, also zu 1a_ideal-matrix-unordered. Der Block unter der obersten Zeile mit den Flächen *Show*, *Click mode*, *Edit* und *View* ist relevant, bei *Click mode* entdecken wir unter dem (wohl noch aktiven) Knopf *Select* den Knopf *Drag row/colums*. Wir klicken auf *Drag row/columns* und aktivieren damit diese Funktion. Nun ist PAST bereit, Zeilen und Spalten in der Datentabelle zu verschieben. Und das tun wir jetzt, wir ordnen durch Verschieben der Zeilen und Spalten die Tabelle neu nach den Ergebnissen der CA, in jener Reihenfolge, die wir aus dem zweiten Fenster Correspondence analysis ablesen. Wir gehen mit dem Zeiger /Cursor auf den Kopf der Spalte Typ-A, klicken und ziehen die Spalte auf die erste Position der Tabelle, nach ganz links. Dann suchen wir die Spalte Typ-B, klicken auf deren Kopf und ziehen die Spalte nach links in die zweite Position der Tabelle, nach Typ-A. Wir fahren fort, und nach ein paar Vertauschungen sind die Typen geordnet in – von links nach rechts – Typ-A, Typ-B, Typ-C usw. Erster Schritt fertig.

Nun zu den Zeilen (Gräbern). Wir können im Fenster *Correspondence analysis* in den Schaltflächen rechts nun die Gräber (*rows*) einblenden und die Typen (*columns*) zur besseren Lesbarkeit ausblenden. Dann sortieren wir in unserer Ausgangstabelle die Gräber (d. h. die Zeilen) in die dort entlang der x-Achse abgelesene Reihenfolge. Wiederum durch Anklicken des Zeilenkopfes links und ziehen der jeweiligen Zeile in die richtige Position. Also erst einmal Grab-1 in die erste, oberste Zeile, dann Grab-2 in die zweite Position, usw. Am Ende sollte

dann die Tabelle mit allen Spalten (Typen) und Zeilen (Gräbern) exakt nach den Ergebnisse der CA geordnet sein (**Abb. 8**).




| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-3 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 |
| grave-8 | 0 | 0 | 0 | 0 | | 0 | 1 | 2 | 1 | 0 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |

**Abb. 8** Bildschirmfoto unserer Beispieltabelle Abb. 4, nun mit den Zeilen und Spalten neu geordnet nach den Ergebnissen der CA.

Vorab: Bei großen Tabellen könnte in der späteren Praxis diese anschauliche und einfach durchzuführende händische Umsortierung der Zeilen und Spalten etwas mühsam werden. Nach Kap. 17 zeige ich daher im Anhang (Kap. 18) einen Weg, bei großen Tabellen schneller zum Ziel zu kommen.

Beachten Sie, dass bei diesem Umordnen nur die Reihenfolge der Spalten und Zeilen verändert wird, nicht der eigentliche Inhalt: Weiterhin befinden sich die Typen in den gleichen Gräbern wie in der Ausgangstabelle. Bitte schauen Sie sich nun die resultierende Tabelle genauer an, um das recht einfache Muster zu sehen, das ich für unsere Übung angenommen habe: In den Spalten sind die Typen immer in-existent (0), werden erfunden (1), sind modern (2), werden wieder unmodern (1) und verschwinden danach gänzlich (0). In den Zeilen haben alle Gräber jeweils drei Typen, zwei von ihnen nur einfach, einen Typen jeweils in zwei Exemplaren. Ja, ein recht einfaches und schematisches Modell, eben eine “ideale Matrix”, zudem mit einer Benennung der Gräber und Typen, die das Ordnen resp. nun die Kontrolle der Ordnung sehr leicht macht. Ganz so einfach wird es später bei realen archäologischen Anwendungen nicht sein, aber für eine erste Erfahrung mit PAST und einer CA dürfte diese Einfachheit hilfreich sein.

## 8 Einige Erläuterungen zu den statistischen Maßzahlen

Zuerst seien die Prioritäten gerade gerückt: Beim Arbeiten mit einer CA sollten nicht die statistischen Werte im Vordergrund stehen, sondern die archäologischen Inhalte und Kriterien. Die Typen oder Merkmale sollten wohlüberlegt und gut definiert sein. Ihre Auswahl muss der Fragestellung angemessen sein und ggf. im Laufe des Arbeitsprozesses optimiert werden. Geht es bei der Analyse um Chronologie, braucht man chronologisch empfindliche Typen; geht es um andere Aspekte wie z. B. das soziale Geschlecht, braucht man Typen, die für

diese Fragestellung relevant und empfindlich sind. Die Gräber (oder Befunde), die in die Analyse eingehen, müssen hinreichend an Zahl und der Fragestellung angemessen sein. Geht es um Chronologie, braucht man gut dokumentierte geschlossene Funde, während Befunde, die eventuell vermischt sind oder über eine lange Zeit hinweg Material gesammelt haben, keine guten Ergebnisse bringen können und aus den Analysen ausgeschlossen werden sollten. Der Fokus des Arbeitens sollte auf der Archäologie liegen, hier liegen sowohl die gravierenderen Fehlerquellen oder als auch die größeren Verbesserungsmöglichkeiten. Ein tieferes Verständnis der nachfolgend erläuterten statistischen Werte ist demgegenüber weniger wichtig, wenn auch nützlich.

### 8.1 Achsen, Eigenwerte und Inertia

Die CA berechnet normalerweise eine mehrdimensionale Lösung: zunächst eine dominante, erste Lösung (oder Ordnung) aus den eingegebenen Daten ("Achse 1"), danach wird eine zweite Lösung gesucht, die unabhängig von der ersten Achse ist, danach eine dritte Lösung, die von der ersten und der zweiten Achse unabhängig ist, und so weiter. Dies ist ein rein statistischer Prozess, der im übrigen ganz ähnlich bei einer PCA/Faktorenanalyse stattfindet, die ebenfalls nacheinander mehrere voneinander unabhängige Faktoren aus einem Datensatz extrahiert. Es ist im Falle einer CA möglich, dass auch die zweite oder gar die dritte Achse eine archäologisch interpretierbare Ordnung aufweist. Aber in der Praxis vieler archäologischen Arbeiten, die eine CA unternommen haben, zeigt sich, dass meist vor allem die erste Achse ergebnisträchtig ist und für die spätere Interpretation genutzt werden kann, dagegen nur selten auch die zweite oder gar dritte Achse. Um ein Beispiel zu geben, wie ein solches mehrdimensionales Ergebnis bei der Analyse von Gräbern im Idealfall aussehen könnte: die erste Achse ordnet nach sozialem Geschlecht, die zweite nach der Chronologie, die dritte nach dem sozialen Status der Verstorbenen. Aber dies ist ein fiktives Modell, eine reale Analyse dieser Art ist mir weder je gelungen noch ist mir eine solche aus der Literatur bekannt. Für unsere Praxis heißt dies, dass wir uns im Normalfall auf die Ordnung nach der ersten Achse fokussieren können und diese optimieren und verstehen sollten.

Alle in der CA berechneten Achsen gemeinsam sollten eine optimale statistische Erklärung der Struktur geben, die in der gesamten Variabilität des eingegebenen Datensatzes steckt. Diese gesamte Variabilität im Datenkörper wird "Inertia" genannt. Die erste Achse spiegelt einen Teil dieser Inertia wider, die zweite Achse einen weiteren, allerdings geringen Anteil usw. Die Bedeutung – im statistischen Sinne – jeder Achse (und Grabes, Befundes, Typs) ist ihr "Eigenwert" (engl. *eigenvalue*). Je höher der Eigenwert einer Achse, desto größer ihre Bedeutung, d. h. ihr Anteil an der gesamten Inertia. In unserem Beispiel hat Achse 1 einen Eigenwert von ca. 0.95 (siehe Spalte *Eigenvalue* hinter dem Reiter *Summary* im Fenster *Correspondence analysis*) oder 30.56 Prozent der Inertia der gesamten Tabelle (siehe Spalte *% of total*). Achse 2 in unserem Beispiel hat einen Eigenwert von 0.80 und einen Anteil von 25.86 Prozent an der gesamten Inertia. Gemeinsam erklären Achse 1 und 2 56.42 Prozent der in der Tabelle steckenden Variabilität (Spalte *Cumulative*) - was im übrigen erfahrungsgemäß

ein recht hoher Wert ist. Normalerweise sollte die erste Achse einen relativ hohen Eigenwert aufweisen und einen hohen Anteil der gesamten Inertia erklären, und die nachfolgenden Achsen sollten eine zunehmend geringer werdende Bedeutung haben.

In der Praxis indes sollten diese Zahlen nicht überbewertet werden. Ich habe Tabellen gesehen, die rein statistisch gesehen sehr gute Werteketten zeigten, aber archäologisch keinen nennenswerten Sinn ergaben, und umgekehrt Tabellen, die archäologisch gut vorbereitet waren und sehr sinnvolle Interpretationen zuließen, jedoch rein statistisch gesehen keine besonders guten Werte aufwiesen. Der archäologische Wert einer CA bemisst sich nicht nach den statistischen Kennzahlen, sondern muss vor allem anhand archäologischer Argumente überprüft und validiert werden (Kap. 12.2).

## 8.2 Zeilen- und Spaltenwerte (*scores*)

Wir verstehen nun auch die Zeilen- und Spaltenwerte (in der älteren deutschsprachigen Literatur auch "Schwerpunkte" genannt). Jede Achse steht für eine eigene (Bedeutungs-)Dimension, und die Zeilen- und Spaltenwerte der einzelnen Gräber (oder Befunde) und Typen (oder Attribute) zeigen deren Position entlang der von der jeweiligen Dimension aufgespannten Skala an. Es sind diese Werte, nach denen die Punkte im Streuungsdiagramm angeordnet werden. Da unsere CA mehr als zwei Dimensionen errechnet hat, gibt es mehr als ein Streuungsdiagramm, das wir betrachten können (Kap. 8.3).

Es ist wichtig zu wissen, dass diese Werte (*scores*) entlang jeder Achse eine gut definierte Skala bilden, auf der jedes Grab und jeder Typ einen bestimmten Platz haben, bei dem der Abstand zum Nachbarn definiert ist und der daher auch interpretiert werden kann. Aber die Skala insgesamt hat keine definierte Richtung; vielmehr kann die Richtung der Achsen nach Belieben umgekehrt werden, z. B. indem man alle Werte mit "minus 1" multipliziert. Eine solche Umkehrung der Richtung verändert nicht die Ordnung und auch nicht die Abstände der einzelnen Punkte zueinander, welche aus den individuellen Werten (*scores*) ersichtlich sind. Um es mit archäologischem Hintergrund einfacher zu erklären: wenn wir eine Achse einer CA als Zeit interpretieren, ist der Wert (*score*) für ein Grab und für einen Typ wohl berechnet und seine Position auf dieser Achse exakt definiert, nicht aber, wo bei der Achse der (alte) Anfang und das (junge) Ende liegt. Diese wichtige Frage kann nie mit Hilfe einer CA gelöst werden. Hierzu bedarf es externer archäologischer Argumente, wie etwa der Stratigrafie oder von $^{14}$C-Daten. Wozu dann die CA? Nun, man benötigt für die Klärung der Richtung eigentlich nur für zwei Fälle einer ganzen Tabelle eine zuverlässige externe Datierungsinformation (Stratigrafie, $^{14}$C, Dendrochronologie, Münzen), und es muss nicht notwendigerweise eine absolute Datierung sein, sondern als Mindestanforderung lediglich eine sichere relative Datierung. Das Übrige leistet dann die CA.

### 8.3 Jenseits der ersten Dimension (Achse) einer CA

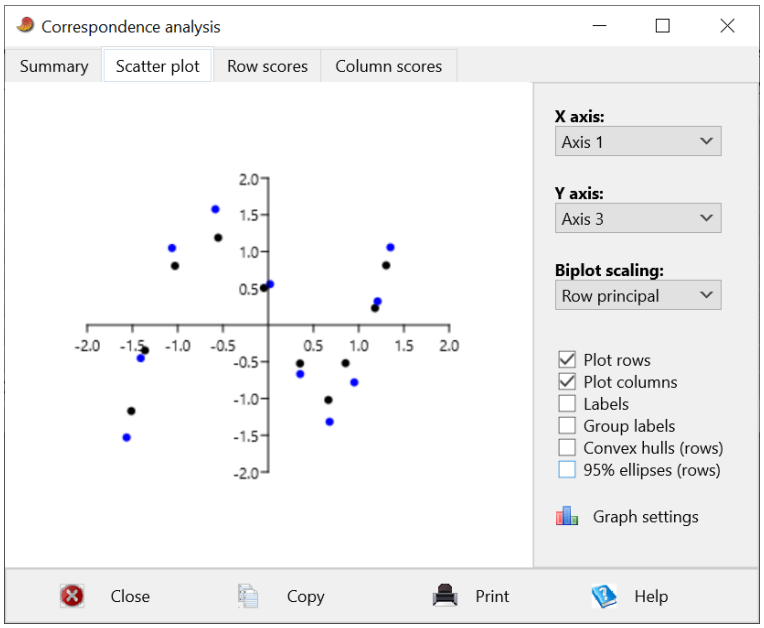
Wie wir oben erfahren haben, errechnet eine CA üblicherweise mehrere Dimensionen bis hin zu dem Punkt, wo aus Sicht der Statistik die weitere Extraktion von Dimensionen keinen Sinn mehr macht. Die Anzahl dieser Dimensionen ist nicht fix, sondern individuell vom Datensatz abhängig. Doch es ist nicht notwendig, sich um all' diese Achsen zu kümmern. In den meisten archäologischen Anwendungen der Methode wird nur die erste, manchmal auch die zweite Dimension interpretiert, selten nur eine dritte oder höhere. In einem Streuungsdiagramm zwischen der ersten und zweiten Achse ergibt die Wolke der Punkte oft die Form einer Parabel oder eines Hufeisens (**Abb. 6**). Diese Figur ist ein rein mathematisches Artefakt des Verfahrens. Wenn die Daten vollständig dem unimodalen Modell folgen und hinreichend dicht und ohne große Lücken die berechneten Dimensionen widerspiegeln, ergibt sich im Streuungsdiagramm von Achse 1 mit Achse 2 solch eine Parabel in einer Idealform. In diesen Idealfällen ergeben auch die Streuungsdiagramme der Achsen 1 und 3 sowie der Achsen 2 und 3 bestimmte ideale Kurvenverläufe. Diese idealen Kurven sollte man einmal gesehen haben und kennen, weshalb wir sie jetzt studieren wollen.

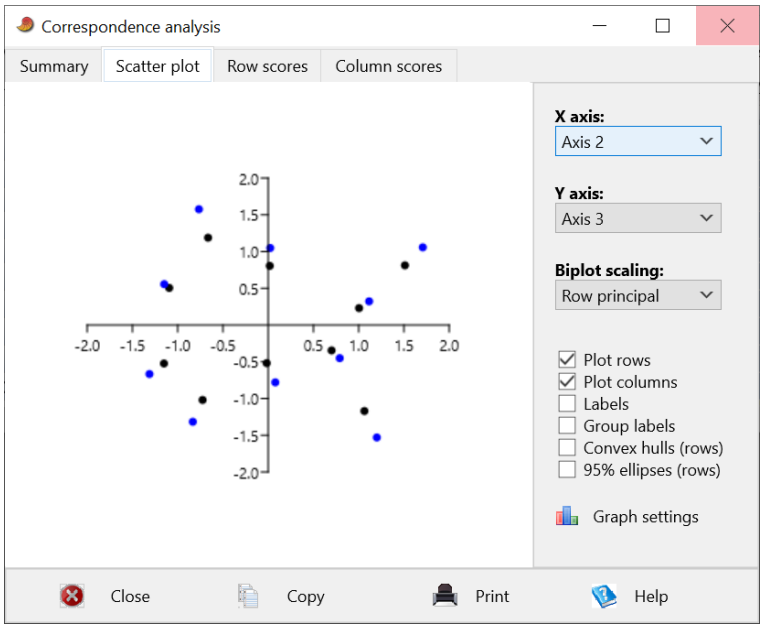
Starten Sie dazu bitte PAST und öffnen die Tabelle 1b_ideal-matrix-ordered und schauen sich die Tabelle noch einmal kurz an (es sind inhaltlich die gleichen Daten wie 1a_ideal-matrix-unordered). Sie zeigt ein modellhaft ideales Bild von zehn Typen und zehn Gräbern mit einer jeweils optimal unimodalen Präsenz der Typen in den Gräbern. Rechnen Sie nun eine CA und schauen sich das Streuungsdiagramm Achse 1 gegen Achse 2 an, so, wie wir es oben praktiziert haben (Kap. 7.1). Sie sollten jetzt eine Parabel sehen, die bestätigt, dass dieser Datensatz in guter Weise dem unimodalen Modell folgt. Im Fenster *Correspondence analysis* befinden sich weitere Schaltknöpfe, die wir bislang noch nicht benutzt haben, nämlich oben jene beiden mit der Überschrift *X axis* und *Y axis.* Gehen Sie nun bitte zur Schaltfläche *Y axis* und klicken auf den Schalter. Es klappt ein Menü auf mit einer Liste der verfügbaren Achsen 1, 2, 3 usw., klicken Sie dort auf Achse 3. Unmittelbar verändert sich das Streuungsdiagramm, es zeigt nun die Achsen 1 (waagerecht, x-Achse) und 3 (senkrecht, y-Achse) (**Abb. 9**). Die Punkte folgen nun einer liegenden S-Kurve, was das erwartete typische Bild ist und erneut bestätigt, dass die Daten dem erwarteten Idealmodell entsprechen. Schalten Sie gegebenenfalls rechts das Häkchen bei *Labels* weg, dann sehen Sie nur die Punkte und können die Kurve besser studieren.

**Abb. 9** Bildschirmfoto des Ausgabefensters von PAST. Es zeigt das Streuungsdiagramm von Achse 1 (waagerecht) gegen Achse 3 (senkrecht). Vgl. Abb. 6 und 10.

**Abb. 10** Bildschirmfoto des Ausgabefensters von PAST. Es zeigt das Streuungsdiagramm von Achse 2 (waagerecht) gegen Achse 3 (senkrecht). Vgl. Abb. 6 und 9.

Wir wollen nun das Diagramm der Achsen 2 und 3 betrachten. Dazu gehen wir auf die Schaltfläche *X axis* und wählen dort die Achse 2 aus, klick. Wieder ändert sich das Streuungsdiagramm, es zeigt jetzt die Punkte im Raum der Achse 2 (waagerecht, x-Achse) und Achse 3 (senkrecht, y-Achse) (**Abb. 10)**. Die Punkte folgen nun – ganz unstatistisch gesprochen


Correspondence analysis
Summary
Scatter plot
Row scores
Column scores
X axis:
Axis 1
Y axis:
Axis 3
Biplot scaling:
Row principal
Plot rows
Plot columns
Labels
Group labels
Convex hulls (rows)
95% ellipses (rows)
Graph settings
Close
Copy
Print
Help


Correspondence analysis
Summary
Scatter plot
Row scores
Column scores
X axis:
Axis 2
Y axis:
Axis 3
Biplot scaling:
Row principal
Plot rows
Plot columns
Labels
Group labels
Convex hulls (rows)
95% ellipses (rows)
Graph settings
Close
Copy
Print
Help

– der Kontur eines von der Seite gesehenen Fisches, was wiederum die erwartete Idealkurve dieser beiden Dimensionen darstellt.

Die drei Streuungsdiagramme von Achse 1 mit 2, 1 mit 3 und 2 mit 3 (Abb. 6, 9-10), die wir hier studiert haben, sind drei zweidimensionale Ansichten auf einen dreidimensionalen Würfel und eine Punktwolke, die innerhalb dieses Würfels in einer komplexen Form verläuft. Wir schauen jeweils von einer Seite des Würfels in diese Wolke hinein. Wenn Sie sich näher für diese Form interessieren, können Sie versuchen, sich ein mechanisches Modell davon zu bauen – doch wirklich nötig ist dies nicht. Für unsere Zwecke sind die drei zweidimensionalen Streuungsdiagramme hinreichend, und es war wichtig, diese idealtypischen Kurvenverläufe zumindest einmal zu sehen. Man sollte sie in Erinnerung haben, wenn mal reale Datensätze untersucht, um die Ähnlichkeit resp. Unähnlichkeit der realen Kurven mit diesen Idealbildern einschätzen zu können.

### 8.4 Korrespondenzanalyse und Seriation

Nun sind wir bereit für die Beantwortung einer sich längst aufdrängenden Frage: Was ist der Unterschied zwischen einer Seriation und einer Korrespondenzanalyse? Nun, wie wir gesehen haben, ergibt die CA eine mehrdimensionale Lösung. Die Seriation tut dies nicht, sie ergibt nur eine Lösung, d. h. sie ist eindimensional. Die Ordnung, welche mit einer Seriation errechnet wird, ist – wenn die Seriation modern und korrekt durchgeführt wurde – identisch mit der ersten Achse einer CA. Daher sind ältere Studien, die anstelle einer damals noch unbekannten CA auf einer Seriation beruhen, heute auch nicht falsch. Vielmehr sollte sich bei Anwendung einer CA auf die damals seriierten Daten eine identische oder zumindest sehr ähnliche Lösung ergeben.

Die erste in Deutschland per Computer durchgeführte Seriation (Goldmann 1972) berücksichtigte nur die Anwesenheit und Abwesenheit von Typen, d. h. die zu ordnende Tabelle bestand nur aus Nullen und Einsen (Anwesenheits- /Abwesenheits-Matrix). Später kam bei der Analyse von Siedlungsmaterial das Bedürfnis auf, auch den Aspekt der unterschiedlichen Häufigkeit von Typen in den einzelnen Inventaren berücksichtigen zu können. Nach einer leichten Anpassung des Rechenwegs (Algorithmus) und einer Verbesserung des statistischen Gütemaßes waren dann seit den 1970er-Jahren auch sog. Häufigkeitsseriationen möglich. Die zu ordnenden Tabellen werden in der archäologischen Literatur meist Kombinationstabellen genannt, unter Statistikern heißen sie Kontingenztafeln. Wendet man den Algorithmus einer Häufigkeitsseriation auf eine Anwesenheits-/Abwesenheits-Tabelle an, ergibt sich die gleiche oder eine sehr ähnliche Ordnung wie bei einer sog. Seriation, allerdings mit besser interpretierbaren “Schwerpunkten” (*scores*).

### 8.5 Was ist relevant: die Kurve oder die Achse?

Nachdem wir die idealtypischen Kurven – insbesondere die Parabel von Achse 1 mit 2 – gesehen haben, schließt sich meist die Frage an, welche Ordnung und welche Position eines Grabes oder eines Typs relevant ist: jene exakt entlang einer Achse (z. B. Achse 1) oder jene entlang des Verlaufs der Parabel? Die erste Antwort ist richtig: Entscheidend sind die Ordnung und die Abstände entlang der relevanten Achse, nicht entlang der Kurve.

Um für die weitere Auswertung einer CA eine bessere Darstellung der Position und der Abstände der Gräber und Typen entlang einer Achse bereitstellen zu können, wurde eine Variante der CA entwickelt, die als *Detrended Correspondence Analysis* (DCA) bezeichnet wird (siehe Kap. 12.5). Dabei wird zunächst eine gewöhnliche CA berechnet, und anschließend werden die resultierenden Kurven aus den Achsen herausgerechnet, so dass diese im Idealfall eine gerade Line bilden. Die Idee dahinter ist, dass nach dem "End-trenden" der Achsen die Abstände zwischen den Punkten richtiger sind als bei einer gewöhnlichen CA. Das mag so sein, aber im Vergleich zeigt sich, dass die Unterschiede zwischen einer CA und einer DCA in der relevanten Achse 1 (und meist auch 2) sehr gering sind und daher – aus meiner Sicht – in den meisten Fällen nicht relevant für die archäologische Praxis.

Neugierig geworden auf eine DCA? Mit PAST nichts leichter als das. Gehen Sie zurück zur Ausgangstabelle mit den Daten, aktivieren Sie wie gewohnt die gesamte Datentabelle oder einen Bereich davon, gehen in die oberste Zeile auf *Multivariate*, » *Ordination*, und nun statt *Correspondence (CA)* eine Zeile tiefer zu *Detrended Correspondence (DCA)*, klick, und fertig. Ein neues Fenster *Detrended correspondence analysis* zeigt ihnen die Ergebnisse, und zwar das Streuungsdiagramm Achse 1 (waagerecht) mit Achse 2 (senkrecht). Wie ein Vergleich mit einer für den gleichen Datensatz berechneten CA schnell zeigt, hat sich die Reihenfolge der Gräber und Typen nicht verändert, sondern allenfalls ein wenig die Abstände und ggf. die Richtung (Sie erinnern sich: Die Richtung kann frei umgekehrt werden).

### 8.6. Der "Parabeltest"

Bisweilen liest man in archäologischen Publikationen, die mit einer CA gearbeitet haben, dass ein Parabeltest durchgeführt wurde. Möglicherweise haben Sie noch nie von diesem Test gehört und konsultieren Ihr gewohntes Handbuch der Statistik, um näheres über den Parabeltest zu erfahren. Dort finden Sie Ausführungen z. B. zum Chi-Quadrat-Test, den Mann-Whitney U-Test oder zum Kruskal-Wallis H-Test, aber leider nichts über den Parabeltest. Kein Wunder, es gibt ihn nämlich nicht. Der "Parabeltest" ist kein seriöses statistisches Testverfahren, sondern ein Mythos. Einen Parabeltest durchführen meint nichts weiter, als rein optisch die tatsächlich beobachtete Streuung der Punkte im Streuungsdiagramm Achse 1 mit Achse 2 mit dem erwarteten Idealbild einer Parabel (wie **Abb. 6**) zu vergleichen. Das ist der "Parabeltest".

Nun, es kann wirklich wertvoll sein zu sehen, inwieweit nach einer CA im Streuungsdiagramm der Achse 1 mit Achse 2 die Punktwolke die Form einer Parabel annimmt, denn dies validiert resp. falsifiziert die zu Grunde liegende Annahme des Vorliegens des unimodalen Modells. Auch die Wahrnehmung der Abweichung von der Idealform einer Parabel kann hilfreiche Hinweise für die weitere Analyse und Interpretation der Daten geben. Aber ein solcher Parabeltest ist alles andere als ein sauberer statistischer Test. Daher: Bitte sprechen Sie nie wieder vom Parabeltest, denn es gibt ihn nicht.

### 8.7. Von Hufeisen und Parabeln

Worum geht es eigentlich: Um eine Parabel oder um eine hufeisenförmige Kurve? Wie wir schon gelernt haben, ist die sich aus einer CA ergebende Ordnung wohldefiniert, nicht aber die Richtung. Die Richtung kann beliebig umgekehrt werden. Oft ergibt das Streuungsdiagramm von Achse 1 mit Achse 2 die Form einer Parabel mit offenen Enden nach oben und einer unten liegenden Mitte, bisweilen aber auch die Form eines Hufeisens mit offenen Enden nach unten und einer oben liegenden Mitte. Die diesbezügliche Lösung der konkreten CA ergibt sich eher zufällig im letzten Schritt der internen Rechnungen, sie ist von der Eingabereihenfolge der Daten und vom Zufall abhängig, und es liegt keine inhaltliche Bedeutung darin. Parabel oder Hufeisen sind daher inhaltlich gänzlich äquivalente Bilder.

Bei der Analyse realer Daten kann es indes hilfreich sein, dem Leser jeweils vergleichbare Bilder anzubieten, d.h. stets eine Parabel oder stets ein Hufeisen, die Achsen also ggf. durch Multiplikation mit “minus 1” entsprechend zu spiegeln. Dabei liegt die Antwort auf die Frage “Parabel oder Hufeisen” ganz bei Ihnen, wie Sie es persönlich bevorzugen. Die Mehrheit der Archäologen in Europa bevorzugt die Darstellung als Parabel, US-amerikanische Archäologen scheinen die Darstellung (und auch die Benennung) als Hufeisen zu bevorzugen. Es ist wichtig zu wissen, dass dies allein die Optik betrifft und es keinen inhaltlichen Unterschied zwischen beiden Traditionen gibt.

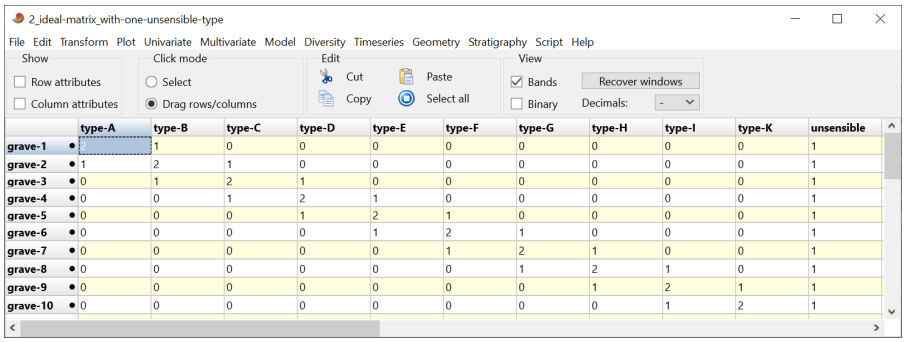
## 9 Mehr Erfahrung gewinnen mit der CA

Nachdem wir an einem Idealfall das Grundlegende kennen gelernt haben und eine CA mit Hilfe z. B. von PAST unfallfrei rechnen können, ist es nützlich, zunächst anhand gezielt konstruierter Daten weitere Erfahrungen mit typischen Fällen und der Interpretation der Ergebnisse zu gewinnen, bevor man sich an echte archäologische Datensätze begibt. Daher sollen hier noch ein paar weitere Beispiele mit kleinen simulierten Datensätzen berechnet und betrachtet werden, um sich etwas tiefer in die Anwendung der CA einzufinden. Dies wird später helfen, echte archäologische Datensätze besser zu verstehen und sie für eine CA gegebenenfalls optimieren zu können.

## 9.1 Fallstudie mit einem unspezifischen Typ

Die Tabelle 2_ideal-matrix_with-one-unspecific-type zeigt idealisiert einen in der archäologischen Praxis häufigen Fall: Die Tabelle enthält vorwiegend gut definierte, zeitempfindliche Typen und gute geschlossene Komplexe resp. Gräber (**Abb. 11**). Ein Typ aber tritt über die ganze hier repräsentierte Zeit hin auf, er ist unspezifisch, zeit-unempfindlich. In der Archäologie wird solch ein Typ oft als "Langläufer" bezeichnet. Um die Dinge hier so einfach wie möglich zu halten, befindet sich die Tabelle bereits in der ideal richtigen Ordnung und am Ende ist jener unsensible Typ der Art "Langläufer" hinzugefügt.

Aktivieren Sie bitte PAST, laden diesen Datensatz (oder geben ihn nach Abb. 11 jetzt ein) und berechnen eine Korrespondenzanalyse. Schauen Sie auf das Streuungsdiagramm Achse 1 mit Achse 2 (**Abb. 12**). Es sieht der ersten, idealen Parabel (**Abb. 6**) ohne Langläufer sehr ähnlich: Das Streuungsdiagramm weist annähernd eine Parabelform auf und die Typen und Gräber liegen weitgehend in der erwarteten Reihenfolge. Neu zeigt das Diagramm den Typ *unsensible*, und zwar in der Mitte der parabelförmigen Streuung der übrigen Punkte. Prima, das ist das typische Bild. Wenn sich bei einer CA mit im allgemeinen gut definierten zeitsensiblen Typen und guten geschlossenen Funden eine stabile Ordnung ergibt und dem entsprechend eine Parabel im Streuungsdiagramm Achse 1 mit Achse 2, liegen die zeit-unempfindlicheren Typen resp. Langläufer gerne inmitten dieser parabelförmigen Streuung. Nun, nachdem wir das Phänomen kennen, können wir es für die vertiefte Analyse einer CA nutzen, und ggf. eben auch für die Überlegung, welche Typen in eine endgültige Analyse eingehen und welche Typen eventuell weniger hilfreich sind und aus der Analyse ausgeschlossen werden könnten oder besser und schärfer definiert werden sollten.

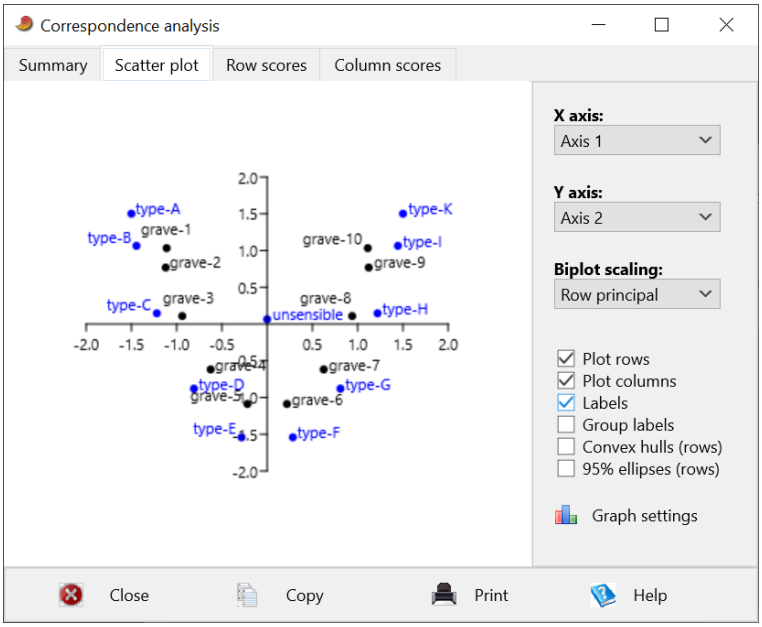



| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K | unsensible |
|---|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| grave-2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| grave-3 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| grave-4 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| grave-5 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 1 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 1 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 1 |
| grave-8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 1 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 1 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |

**Abb. 11** Die Modelltabelle mit einem zusätzlichen, (zeit-)unspezifischen Typ (Spalte rechts).

**Abb. 12** Streuungsdiagramm von Achse 1 (waagerecht) gegen Achse 2 (senkrecht) nach der CA der Tabelle Abb. 11 mit einem unspezifischen Typ.

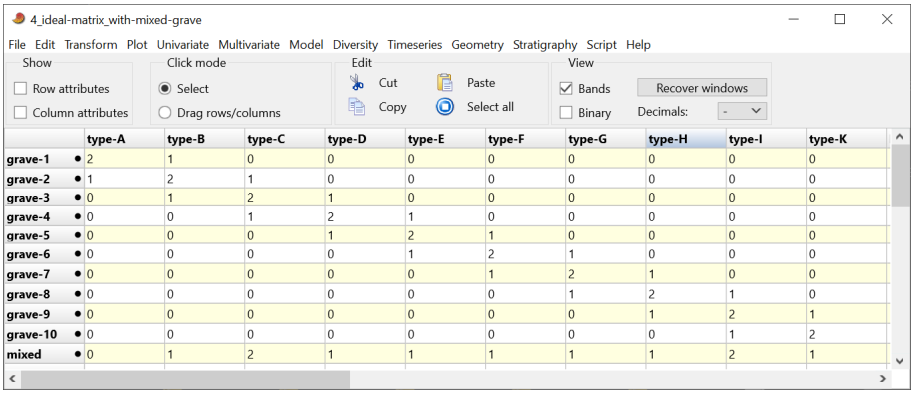
9.2 Fallstudie mit einem vermischten Grabinventar. Man achte in der Mitte des Koordinatensystems auf den Typ "unsensible".




Ähnliches wie beim Fall zuvor ist auch auf Seite der Befunde denkbar, also ein Grab, das über eine längere Zeit Funde "gesammelt" hat. In dem Datensatz 3_ideal-matrix_with-unspecific-grave ist dieser Fall simuliert. Laden Sie die Tabelle, schauen sich das neu eingefügte unspezifische Grab an und führen eine CA durch. Das resultierende Streuungsdiagramm sieht dem vorherigen Bild (**Abb. 12**) sehr ähnlich: Die Typen und Gräber sind entlang einer parabelförmigen Punktwolke gut und richtig geordnet, in der Mitte liegt der neue Befund *collectorgrave*. Die beiden simulierten Fälle zeigen, dass die CA gleicherweise auf Änderungen bei den Befunden (Zeilen) und bei den Gräbern (Spalten) reagiert und dass die besondere Lage von Punkten in der Mitte zwischen den Parabelästen in beiden Fällen ein nützliches diagnostisches Instrument ist.

Lassen Sie uns den zuvor simulierten Fall ein wenig dramatisieren, indem wir ein vermischtes Grabinventar simulieren, das versuchsweise aus einem der älteren und einem der jüngeren Gräber zusammengesetzt wird. Der Datensatz 4_ideal-matrix_with-mixed-grave enthält diesen Fall eines Inventars, das aus Grab 3 und Grab 9 zusammengesetzt wurde, wobei in der Mitte weitere Typen aufgefüllt sind (**Abb. 13**). Nach der CA zeigt sich im Streuungsdiagramm Achse 1 mit Achse 2 das nunmehr bereits erwartete Bild einer weiterhin brauchbaren Ordnung, bei der das Grab *mixed* inmitten der Parabel landet (**Abb. 14**). Im Gegensatz zu den beiden vorherigen Versuchen ist jedoch die Parabel ein wenig verändert: unsymmetrisch und im Falle der Typen H, I und K auch nicht mehr exakt in der von uns vorgesehenen Reihenfolge. Wir lernen: Bei den beiden Versuchen zuvor mit jeweils einem unspezifischen Typ (Langläufer) oder Inventar wurde die wichtige Regel "unimodales Verhalten" nicht schwerwiegend verletzt. Im vorliegenden Fall jedoch zeigen sich zwei voneinander entfernte Typ-Maxima, d. h. das Grab verhält sich letztlich bimodal – und verletzt damit die Modellerwartungen einer CA

gravierender. Ein solcher Fall hat offensichtlich stärkere Auswirkungen auf die Ergebnisse einer CA und die resultierende Ordnung.




| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-3 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 |
| grave-8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| mixed | 0 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |

**Abb. 13** Die Modelltabelle mit einem zusätzlichen, vermischten Grabinventar (unterste Zeile).

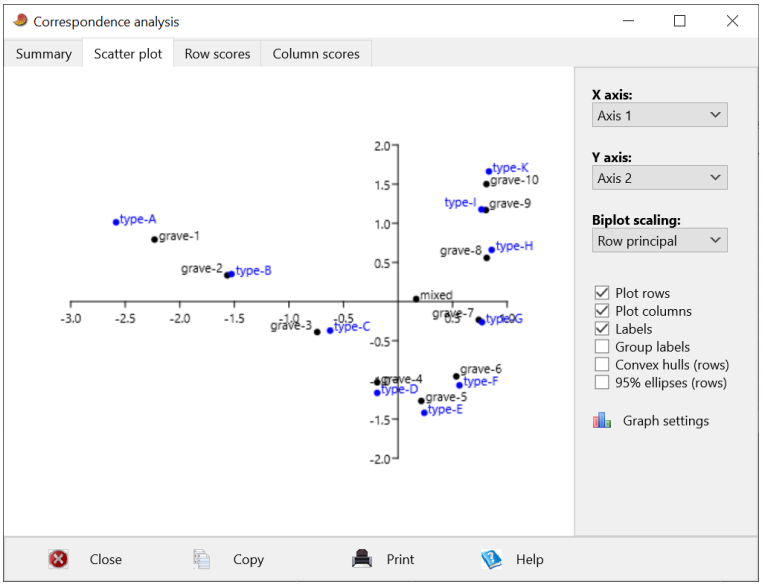


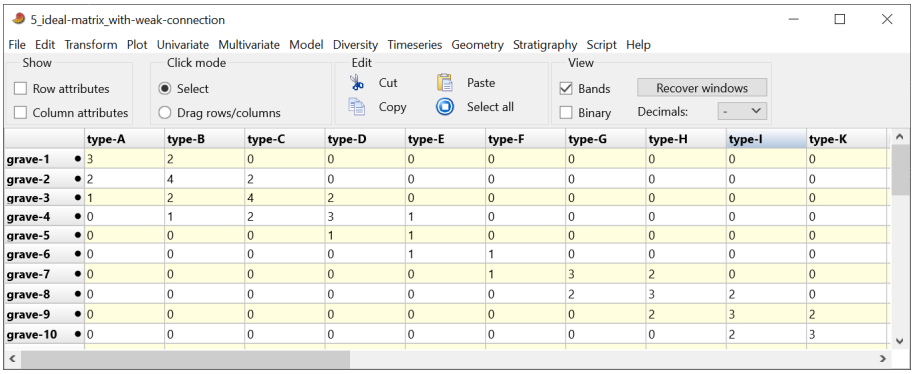

**Abb. 14** Streuungsdiagramm von Achse 1 (waagerecht) gegen Achse 2 (senkrecht) nach der CA der Tabelle Abb. 13 mit einem vermischten Grabinventar. Man achte ca. in der Mitte des Koordinatensystems auf dedas Grab "mixed".

Im Vergleich zu realen archäologischen Datensätzen ist unsere Übungstabelle klein und daher auch empfindlich gegen einzelne Änderungen. Man kann sich diese Erfahrung spielerisch selbst erarbeiten – etwa in dem man die vorgegebene Tabelle 1a_ideal-matrix-ordered zunächst im Sinne ihrer Ausgangsidee deutlich vergrößert und dann die hier exemplarisch durchgeführten Simulationsversuche mit einem einzelnen hinzugefügten, unpassenden Spezialfall wiederholt.

Große, echte Datensätze lassen sich kaum durch solche Einzelfälle gravierend beeinflussen. Aber die hier vorgeführten Beispiele zeigen, dass und wie einzelne Verletzungen der Modellannahmen auf das Gesamtergebnis einwirken. Typen oder Gräber, die im Vergleich zum übrigen Material in einem Datensatz weniger zeitempfindlich sind, stören das resultierende Gesamtbild kaum. Die CA resultiert weiterhin in einer generell stimmigen Ordnung, aber die betreffenden Störfälle werden nicht wirklich sinnvoll in das Bild eingefügt. Immerhin hilft die CA, diese Störfälle aufzuspüren (**Abb. 12, 14**). Tatsächlich vermischte Befunde (Gräber), die typische Ensembles von zwei deutlich unterschiedlichen Zeiten vereinen, sind jedoch einflussreicher. In unserem Fall hilft auch hier die CA, diesen Störfall zuverlässig zu entdecken (**Abb. 14**). Wenn solche Fälle innerhalb eines Datensatzes selten bleiben, ergibt die CA dennoch insgesamt ein brauchbares Bild und hilft außerdem, diese Störfälle zu identifizieren. Werden solche Fälle jedoch in einer Tabelle häufiger, wird es auch mit Hilfe einer CA schwer werden, eine gute Ordnung zu gewinnen und diese störenden Einzelfälle zuverlässig zu identifizieren.

### 9.3 Fallstudie schwach verbundene Datensätze

In den hier angenommenen archäologischen Fällen untersucht die CA die Kombination von Typen in Gräbern. Ein Typ, der nur in einem Grab vorkommt, ergibt keine Fundkombination, ebensowenig ein Grab, das nur einen Typ beinhaltet; beide Fälle tragen nicht zum Untersuchungsgegenstand Fundkombination bei und sollten daher gar nicht erst in die Datentabelle aufgenommen werden. Die Mindestanforderung an eine Untersuchung lautet: Jedes Grab enthält mindestens zwei relevante Typen, und jeder Typ kommt in mindestens zwei relevanten Gräbern vor. Aber auch dann, wenn diese Regel befolgt wird, kann es in umfangreicheren Tabellen Bereiche geben, die vergleichsweise schwach besetzt sind, d. h. die (zu) wenige analysierbare Kombinationen enthalten. Wir wollen uns auch diese Möglichkeit in der Praxis anschauen und laden dazu den Datensatz 5_ideal-matrix_with-weak-connection (**Abb. 15**), schauen uns die Tabelle sorgfältig an und rechnen eine CA. Im Vergleich zu den bisher untersuchten Tabellen weist dieser Modellfall nun höhere Typ-Häufigkeiten an beiden Enden der Tabelle auf, ist aber in der Mitte deutlich ausgedünnt: Schauen Sie insbesondere auf die Gräber 5 und 6 sowie die Typen E und F. Dennoch ist die oben genannte Mindestanforderung "jeder Typ in zwei Gräbern, jedes Grab enthält zwei Typen" weiterhin erfüllt.




| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-2 | 2 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-3 | 1 | 2 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 1 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 0 | 0 |
| grave-8 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2 | 0 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 |

**Abb. 15** Veränderte Modelltabelle mit Gräbern und Typen, die stärker untereinander verknüpft sind, jedoch eine nur schwach verbundene Mittelzone aufweisen.

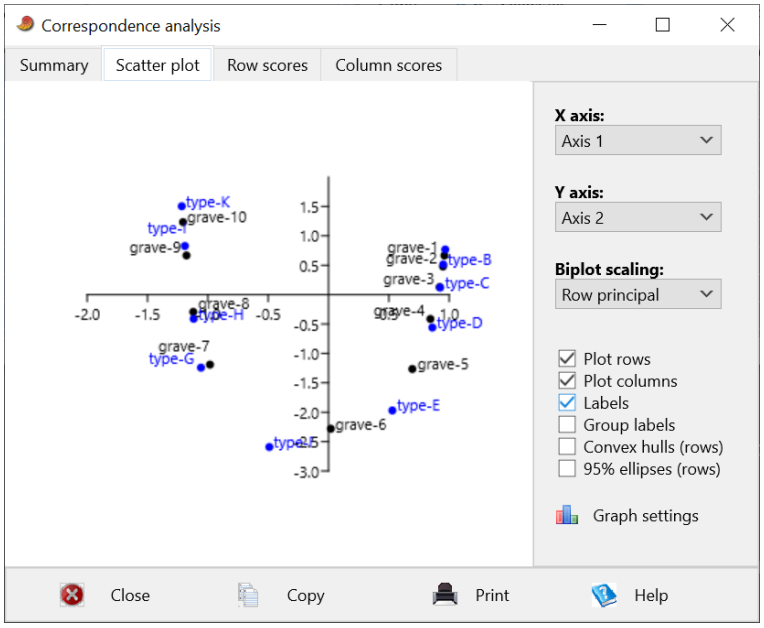



**Abb. 16** Streuungsdiagramm von Achse 1 (waagerecht) gegen Achse 2 (senkrecht) nach der CA der Tabelle Abb. 15 mit einer schwach verbundenen Mittelzone.

Das aus der CA dieser Tabelle resultierende Streuungsdiagramm Achse 1 mit 2 (**Abb. 16**) spiegelt diese Eigenheiten unserer Modellannahmen gut wider: Generell liegen die Gräber und Typen entlang Achse 1 in der erwarteten Ordnung. Die Typen A, B, C und D einerseits und die Typen G, H, I und K andererseits liegen jedoch recht dicht beieinander an jeweils einem Ende der Achse. Bei den Typen A, B und C hatten wir die Typhäufigkeit im Maximum auf 4 erhöht, entsprechend dichter liegen sie entlang Achse 1. Demgegenüber war die maximale Typhäufigkeit bei den Typen G, H, I und K nur auf 3 gesetzt worden, im Streuungsdiagramm liegen sie dicht, aber weniger dicht als A, B und C beieinander. In der Mitte weist die Parabel **Abb. 16** recht große Abstände zwischen den Gräbern resp. Typen auf und ist dort dünn besetzt – was ziemlich genau unseren modellhaften Eingriffen in die Ausgangstabelle entspricht.

Was lehrt das Beispiel? Gräber und Typen können stärker und schwächer als üblich miteinander vernetzt sein. Das Streuungsdiagramm der CA spiegelt diese Verdichtungen und Verdünnungen, d. h. mehr oder weniger starken Vernetzungen, angemessen wider. Man kann dies für die spätere Chronologie z. B. für eine Phasenbildung benutzen, d. h. die Grenzen von chronologischen Phasen gezielt in die dünn besetzten Zonen der Parabel bzw. von Achse 1 legen und sie damit optimal an das gegebene Material anpassen. Es ist sicher nicht notwendig, sich für eine Phasenbildung an solche Kriterien zu halten, nämlich dann nicht, wenn es für die Phasengliederung andere gute Argumente gibt. Fehlen indes solche anderen externen Argumente, kann das Ergebnis der CA hilfreich herangezogen werden. Im vorliegenden Fall würde

man – dem größten Abstand entlang Achse 1 folgend – zwischen die Typen E und F eine Phasengrenze legen und entsprechend zwischen die Gräber 6 und 7.

Unser hier verfolgtes Beispiel weist eine relativ schwache Verbindung zwischen zwei stärker verbundenen Teilbereichen einer Tabelle aus. Fehlt die Verbindung zwischen Blöcken gänzlich oder nahezu gänzlich, kann dies zu schwerwiegenden Fehlern in der per CA gewonnenen Abfolge der Gräber und Typen führen (Ihm 1983, 19-20) – mit einer simulierten Tabelle lässt sich dies leicht ausprobieren. Daher sollte die Tabelle bei einer Analyse realer Daten sorgfältig auf dieses Problem hin überprüft werden.

### 9.4 Die Tabelle ist wichtiger als das Streuungsdiagramm

Wir haben nun mehrere Experimente mit unterschiedlichen künstlichen Daten durchgeführt, denen jeweils konkrete modellhafte Annahmen zu Grunde lagen. Dabei haben wir die Erfahrung machen können, dass das Streuungsdiagramm Achse 1 mit 2 recht unmittelbar wertvolle Einsichten in die Eigenheiten und die Struktur der Daten gewährt. Dennoch ist das Streuungsdiagramm der beiden Eigenvektoren 1 und 2 kein Selbstzweck oder sollte in seiner Bedeutung nicht überschätzt werden. Im Kern geht es um die eingegebene Datentabelle, resp. um jene Tabelle, die wir nach der Umordnung durch die CA erhalten, denn nur hier lassen sich die tatsächlichen Fundkombinationen beobachten und vertiefend analysieren und bewerten. Dies ist auch deshalb wichtig, weil am Beginn einer Analyse in größeren Tabellen erfahrungsgemäß noch Datenfehler stecken, bisweilen schlichte Tipp- oder Übertragungsfehler. Nur beim Blick auf die Tabelle können sie entdeckt und bereinigt werden.

Ein typisches Problem beim Erarbeiten einer Typologie, die ja der CA zu Grunde liegt, ergibt sich aus dem Bedürfnis oder Anspruch vieler Bearbeiter, ein Fundmaterial nach Möglichkeit vollständig, d. h. ohne Rest zu gliedern. Üblicherweise bleiben nach dem Erarbeiten einer ersten Typologie und deren Anwendung auf die Objekte einige wenige Stücke übrig, die zu keinem der bereits definierten und gut mit Objekten unterfütterten Typen wirklich passen. Man wird dann dazu neigen, solche Objekte gegen Ende der Typisierungsarbeit als "untypische" Vertreter jener Objektgruppe zuzuordnen, zu der sie noch am ehesten zu passen scheinen. Bei einem insgesamt guten Material und einer guten Typologie werden viele dieser typologischen ad-hoc-Entscheidungen nach einer Ordnung der Tabelle mit Hilfe einer CA nicht weiter auffallen – weil es sich um gute Entscheidungen eines erfahrenen Materialbearbeiters handelte. Einige dieser ad-hoc-Entscheidungen werden jedoch später dank der CA wieder emporgespült: An der geordneten Tabelle fallen manche dieser Objekte als einzelne, weit von der Diagonale entfernt liegenden "Ausreißer" auf. Es ist wichtig, nach Erreichen einer ersten sinnvollen Ordnung der Tabelle all' diese "Ausreißer" Zeile für Zeile und Spalte für Spalte sorgfältig zu studieren und zu verifizieren. Oft handelte es sich um Einzelfälle, d. h. um tatsächlich ungewöhnliche Fundkombinationen. Nicht selten aber handelt es sich auch schlicht um Fehler oder eben um jene vereinzelten typologischen Entscheidungen, mit denen man als Bearbeiter von Anbeginn an nicht wirklich glücklich war und die sich nun als ungeeignet erweisen. Die

anhand der CA geordnete Tabelle kann helfen, solche den Daten zu Grunde liegenden typologischen Entscheidungen noch einmal zu überdenken – aber eben vor allem anhand der Tabelle und weniger anhand der Streuungsdiagramme.

## 10 Zwei Beispiele von echten archäologischen Datensätzen

Bis hierhin haben wir aus guten Gründen mit künstlichen Datensätzen gearbeitet, die nach explizit formulierten Modellvorstellungen konstruiert worden waren. Nun wollen wir zwei etwas größere reale archäologische Datensätze anschauen, die der Literatur entnommen sind und an denen wertvolle Einsichten gewonnen werden können: zunächst der Datensatz Langweiler-2_Stehli-1973-p-91-fig49 und dann die Tabelle Schretzheim-beads_Koch-U-1977-table-4, womit in den Benennungen zugleich die Quellen dargelegt sind.

### 10.1 Stehli (1973): verzierte Keramik aus einer frühneolithischen Siedlung

Der erste Datensatz stammt aus der Analyse des Siedlungsplatzes Langweiler 2 im Rheinland, einem Fundplatz der sog. Linienbandkeramik (ca. 5.500 - 4.900 v. Chr.), der ältesten neolithischen Kultur in Westdeutschland (Stehli 1973, 91 Abb. 49). Die Tabelle zeigt in den Zeilen als Untersuchungseinheiten ("geschlossene Funde") die Siedlungsgruben aus dieser Siedlung und in den Spalten als Typen (Namensschema a00) die charakteristischen Verzierungen der damals üblichen Feinkeramik. Das hier herangezogene Beispiel ist heute durch jüngere Arbeiten, die ein weitaus umfangreicheres Fundmaterial heranziehen konnten, überholt, aber nach meiner Kenntnis ist diese Studie von Stehli (1973) der erste Fall einer Anwendung der Seriation auf archäologisches Fundmaterial, bei der statt einer Anwesenheits-/Abwesenheitstabelle der Aspekt der Häufigkeit der Typen mitberechnet wurde. In der Originalpublikation wurden die Gruben als Ergebnis dieser Seriation in drei Perioden (1-3) unterteilt. In der hier verwendeten Tabelle sind diese Periodenziffern den betreffenden Bezeichnungen der Gruben vorangestellt.

Unser Datensatz ergibt nach Durchführung einer CA weitgehend jene Ordnung, welche Petar Stehli (1973) bereits aufgrund seiner Seriation erarbeitet hatte. Die Abfolge der von Stehli umrissenen drei Phasen 1 bis 3 wird im wesentlichen verifiziert. Im Detail allerdings unterscheidet sich das Ergebnis unserer CA von Stehlis Ergebnissen. Blicken wir zunächst auf das Streuungsdiagramm. Hier ist die erwartete Parabel weitaus weniger deutlich ausgeprägt als in den bisher verfolgten künstlichen Datensätzen. Das Bild hier entspricht nach meinen Erfahrungen gerade bei Siedlungsmaterial weitaus mehr der archäologischen Realität (z. B. Siegmund 2013, Abb. 2-3 u. 5). Man könnte das Streuungsdiagramm Achse 1 mit 2 als Hinweis darauf lesen, dass die Befunde 1-0485 und 2-0821 Typen unterschiedlicher Zeitstellung vermischen und dass der Verzierungstyp a12 nicht sehr zeitsensibel ist, sondern eher ein Durchläufer. Doch dies sind nur erste Hinweise, die bei einer echten Studie nun anhand des archäologischen Materials eingehend untersucht und verifiziert oder falsifiziert werden müssten. Jedenfalls wird die Ordnung besser – in einem rein technischen Sinn – wenn man versuchsweise die beiden genannten

Inventare und den Verzierungstyp a12 aus dem Datensatz löscht. Versuchen Sie es (markieren, *Edit*, » *Remove…*).

## 10.2 Koch (1977): frühmittelalterliche Perlenketten

Unser zweites Beispiel stammt aus der Monografie von Ursula Koch (1977), in der sie u. a. die Perlen und Perlenketten des frühmittelalterlichen Gräberfeldes bei Schretzheim (ca. 530 - 665 n. Chr.) in Süddeutschland untersucht. Solche Perlenketten sind ein häufiger Bestandteil der frühmittelalterlichen Frauentracht; sie wurden als Kette um den Hals getragen oder auch als Anhänger, z. B. am Gürtel. Als Beigaben an die Verstorbenen gelangten sie in deren Gräber. Koch hat damals die verzierten Perlen untersucht, sorgsam typisiert und dann die Kombination charakteristischer Typen in den als geschlossene Funde betrachteten Ketten untersucht. Ihre originale Tabelle (Koch 1977, Taf. 4) führt die Typen in den Spalten auf und die Gräber resp. die Perlenketten in den Zeilen, die Zellen geben deren Häufigkeit an. Eine Kopie der gedruckten Tabelle ist in unseren Beispieldatensätzen in Form einer MS-Excel-Tabelle enthalten (9_Koch-U-1977-table-4_xls-format; vgl. Koch 1977, Taf. 4); sie entspricht der publizierten Tafel und versucht, deren Druckbild weitgehend zu imitieren. In der damaligen Publikation war die Tabelle ohne Statistik allein durch eine händische Ordnung des Materials seitens U. Koch entstanden. Wie an einem mit Zahlen gefüllten Dreieck und einem leeren Dreieck der rechteckigen Tabelle deutlich sichtbar ist, folgt Koch einem speziellen Konzept von Chronologie: "das jüngste Stück datiert die Kette" resp. das Grab. Dies ist ein häufig gedachtes Modell z. B. auch in der Numismatik, wenn dort Horte resp. Münzschatzfunde untersucht werden. Wir werden auf den methodischen Aspekt im anschließenden Kap. 11 noch einmal zurückkommen. In unserer zum direkten Einlesen nach PAST vorbereiteten Fassung der Tabelle (7_Schretzheim-beads-Koch-1977-table-4) sind die Gräber resp. Perlenketten (Zeilen) in einer speziellen Weise kodiert: die erste Zahl gibt jene Phase der chronologischen Ordnung des Gräberfeldes an, die sich aus der aktuellen Chronologie des Gräberfeldes von Schretzheim ergibt (Koch 2004), es folgt ein Bindestrich und dann die Grabnummer wie in der Originaltabelle (Koch 1977, Taf. 4). Undatierte Gräber zeigen statt einer Zahl ein "x" als führendes Zeichen. Mit Hilfe dieser Kodierungstechnik können wir die Ergebnisse der CA schneller und einfacher mit der konventionellen Datierung der Gräber vergleichen und erkennen, inwieweit die Ordnung nach der CA der zeitlichen Abfolge der Gräber entspricht.

Wenn man eine CA dieser Tabelle rechnet (was Sie jetzt tun sollten), wird schnell deutlich, dass die gewonnene Ordnung der Perlenketten recht gut mit den Datierungen der Grabinventare in Schretzheim übereinstimmt. Nur im Detail erkennt man kleine Unterschiede. Beginnen wir mit dem Befund, dass das Streuungsdiagramm Achse 1 mit 2 eine Parabel ergibt, die erheblich besser ausgeprägt ist als beim vorangehenden Beispiel der bandkeramischen Siedlung Langweiler 2. Mit inzwischen geübtem Blick erkennen wir in der Mitte des Parabelbogens einige Spezialfälle: Grab 6-258 und 7-420 und wohl auch den Perlentyp 33,15-16. Weil hier die Zeit fehlt, tiefer in die archäologische Diskussion dieses Befunds einzusteigen, wählen wir die schnelle einfache Lösung und löschen die betreffenden beiden Zeilen und die Spalte aus dem

Datensatz (markieren, *Edit*, » *Remove*). Berechnen Sie erneut eine CA und vergleichen Sie die Ergebnisse. Die nun gewonnene Ordnung der Perlentypen spiegelt die Datierung der Schretzheimer Gräber besser als zuvor wider und würde eine gute Phasengliederung des verzierten Perlenmaterials in die Schretzheim-Phasen 5, 6 und 7 ermöglichen.

## 11 "Der jüngste Typ datiert den Komplex" - oder: was datiert die CA?

Wie zuvor erwähnt, folgen z. B. viele Numismatiker bei der Analyse von Münzschätzen dem Konzept "das jüngste Stück datiert den Komplex". Überträgt man dieses theoretische Modell in eine konkrete Tabelle, sollte sie ähnlich aussehen wie die zuvor besprochene Tabelle für die Perlenketten aus Schretzheim (Koch 1977, Taf. 4): eine rechteckige Tabelle weist ein leeres Dreieck auf und ein mit Zahlen gefülltes Dreieck, wobei entlang der Diagonalen die Zahlen dichter liegen und im Idealfall dort auch die höheren Häufigkeiten angeordnet sind. Dieses Bild unterscheidet sich von einer Tabelle, die mit Hilfe einer CA geordnet wurde, weil bei einer CA die Häufigkeiten entlang der Diagonalen konzentriert sind, und zwar annähernd symmetrisch mit Zahlen oberhalb und unterhalb der Diagonalen. Das dahinter stehende Modell beinhaltet die Idee, dass die CA den mittleren Zeitpunkt der Grabinventare schätzt und den Mittelpunkt der Verwendungsspanne von Typen – und nicht ihr jüngstes Auftreten. Grabinventare (wie auch Siedlungsgruben) sind Ensembles von Typen: Manche Stücke wird der Tote bereits in seiner Jugend erworben und lange benutzt haben, manche Stücke erst in höherem Alter, und manche Objekte mögen dem Toten eventuell erst zum Zeitpunkt der Bestattung ins Grab gegeben worden sein. Alle Objekte wurden bei der Beerdigung gemeinsam gleichzeitig deponiert, sie können zu diesem Zeitpunkt aber unterschiedlich alt gewesen sein. Die CA schätzt letztlich nicht den Zeitpunkt der Bestattung, sondern das wahrscheinliche mittlere Alter des gesamten Ensembles und entsprechend das wahrscheinliche mittlere Alter der Typen. Wenn Ihnen dieses Modell einleuchtet und zusagt, ist die CA für Sie die Methode der Wahl. Wem dieses Modell misshagt, sollte keine CA verwenden.

Möglicherweise stellt sich jetzt die Frage, welches multivariate Verfahren denn zu dem Modell "das jüngste Stück datiert den Komplex" passt. Fehlanzeige: es gibt kein passendes seriöses multivariates Verfahren für dieses Modell. Denkt man versuchsweise etwa an ein lineares multivariates Verfahren wie etwa eine Hauptkomponentenanalyse (PCA; Kap. 12.5), zeigt ein einschlägiger Versuch z. B. mit dem Datensatz zu den Perlen aus Schretzheim recht schnell, dass die daraus resultierende Ordnung die Chronologie der Gräber erheblich schlechter widerspiegelt als die CA. Machen Sie den Versuch, PAST enthält die nötigen Werkzeuge. Persönlich halte ich den Mangel an einem geeigneten statistischen Verfahren für verschmerzbar, denn aus meiner Sicht eignet sich das Modell "das jüngste Stück datiert den Komplex" nicht für archäologische Probleme.

## 12 Start in eigene Projekte

Nach Durcharbeiten dieser Kapitel sind Sie reif für den Start mit eigenen Fragestellungen und echten archäologischen Daten. Die folgenden Abschnitte möchten Ihnen den Start in ein eigenes Projekt mit ein paar Hinweisen und Ratschlägen erleichtern.

### 12.1 Datenvorbereitung, oder: wie sieht eigentlich die richtige Tabelle aus?

Diese Frage ist weniger seltsam als sie zunächst klingt. Normalerweise sind die Informationen in etwa in folgender Weise verfügbar: Grab 1 enthält ein Schwert vom Typ 1 und einen Schild vom Typ 44; Grab 2 enthält ein Schwert vom Typ 2 und einen Schild vom Typ 55. Wenn man solche Informationen in eine Tabelle umsetzt, könnte diese Tabelle in etwa wie folgt aussehen (**Abb. 17**):

| | Sax | Schild |
|---|---|---|
| **Grab 1** | Typ 1 | Typ 44 |
| **Grab 2** | Typ 2 | Typ 55 |

**Abb. 17** Einfache Tabelle mit den im Text genannten Informationen zu den Typen in zwei Gräbern.

Aber diese Art einer Tabelle ist nicht geeignet für eine CA. Folglich müssen die Informationen in eine geeignete Darstellung übertragen werden (**Abb. 18**):

| | **Sax Typ 1** | **Sax Typ 2** | **Schild Typ 44** | **Schild Typ 55** |
|---|---|---|---|---|
| **Grab 1** | 1 | 0 | 1 | 0 |
| **Grab 2** | 0 | 1 | 0 | 1 |

**Abb. 18** Gegenüber Abb. 17 veränderte Tabelle, die als Eingabe für eine CA geeignet ist.

In der veränderten Tabelle **Abb. 18** steht jede Zeile für ein Grab und jede Spalte für einen Typ. Die Zellen der Tabelle zeigen die Häufigkeit, mit der ein Typ in einem Grab vorkommt, in unserem Fall jeweils eine Null oder eine Eins. Es ist wichtig, den Unterschied zwischen den beiden Tabellen **Abb. 17** und **Abb. 18** zu verstehen und die vorliegenden Daten im Sinne der Tabelle **Abb. 18** aufzubereiten.

### Symmetrische Tabellen / Burt-Tabellen

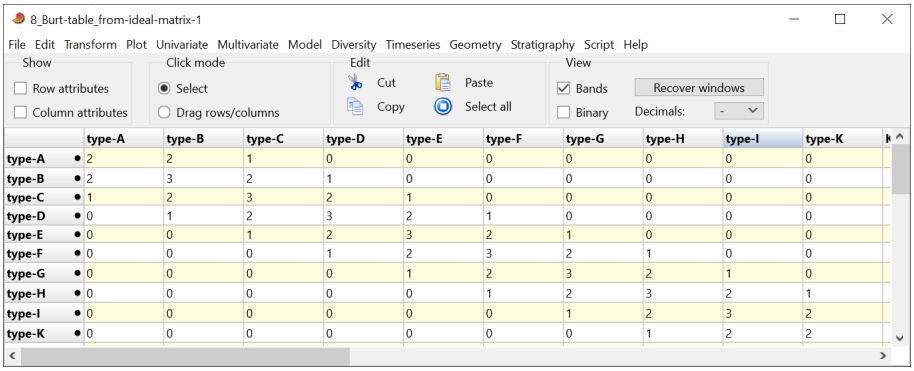
Es gibt eine weitere Art von Tabellen, die in der älteren archäologischen Literatur vorkommt, aber für eine CA nicht geeignet ist. Es handelt sich um quadratische symmetrische Tabellen, die in den Zeilen ebenso wie in den Spalten Typen zeigen und die in den Zellen ausweisen, wie oft dieser Typ in einem geschlossenen Fund mit einem anderen Typ kombiniert vorkommt. Bei derartigen Tabellen sind die Werte spiegelsymmetrisch um ihre Diagonale angeordnet, weshalb sie gelegentlich auch nur als Dreieck dargestellt werden. In der Diagonalen ist ausgewiesen, wie oft ein Typ mit sich selbst kombiniert vorkommt; gelegentlich wird auf diese Information auch verzichtet und die betreffenden Zellen bleiben leer. Heutzutage werden diese Tabellen in der statistischen Literatur als "Burt-Tabellen" bezeichnet. In den Beispieldatensätzen habe ich den Inhalt unserer Tabelle 1a_ideal-matrix-ordered in eine solche Tabelle übertragen und als 8_Burt-table_from-ideal-matrix-1 abgelegt (**Abb. 19**). Soweit mir bekannt, geschah die erste Verwendung einer Burt-Tabelle in der Archäologie durch Heinz Gatermann (1942, 11 Abb. 1), der in seiner Dissertation die Verzierung von Keramik der Becherkulturen in Westdeutschland untersuchte und sie zusammenfassend in einer Tabelle dieser Art darstellte. Seine Studie inspirierte David L. Clarke (1970, 429 u. 469), solche Tabellen auch in seiner Arbeit über die becherzeitliche Keramik in Großbritannien und Irland einzusetzen. Weitere Beispiele dieser Tabellen in archäologischen Arbeiten finden sich z. B. bei Neuffer (1965) und Gebühr (1970).

Tabellen dieser Art sollten nicht mit einer CA analysiert werden, denn obwohl dies rein technisch möglich ist, wären die Ergebnisse nicht korrekt. Greenacre (2007, 137-152) erläutert die statistischen Probleme eines solchen Vorgehens ausführlich und skizziert eine mögliche Lösung, die er *Joint Correspondence Analysis* (JCA) nennt. Aber eine JCA erfordert andere Berechnungswege. (Dieses Verfahren ist im Methodenspektrum von PAST nicht enthalten). Auch von einem archäologischen Standpunkt aus gesehen spricht manches gegen die Verwendung dieser Tabellen, weil die zu Grunde liegenden Informationen in ihnen nicht mehr sichtbar sind: nämlich die konkreten Fundkombination in den Gräbern. Arbeitet man wie oben beschrieben an den Daten und möchte z. B. Typisierungen überarbeiten oder ungeeignete Gräber resp. Typen aus der Tabelle entfernen, ist dies bei Burt-Tabellen ein recht komplizierter Prozess. Obwohl die Methode der JCA einen Ausweg böte und solche Tabellen berechenbar machen würde, empfehle ich daher sehr, von der Verwendung von Burt-Tabellen abzusehen.

**Abb. 19**. Die Daten aus der idealen Tabelle (Abb. 3), umgewandelt in eine "Burt-Tabelle", die nicht als Eingabe für eine CA geeignet ist.

## 12.2 Man braucht gutes Material, eine gute Fragestellung und eine geeignete Prüfhypothese

"Um eine sichere Chronologie für die vorgeschichtlichen Zeiten zu erhalten, muss man ein grosses Material und eine gute Methode haben" schrieb Oscar Montelius 1903 in seinem wichtigen Buch "Die Methode" (Montelius 1903, 2; Hervorhebungen O. Montelius). Und




| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| type-A | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| type-B | 2 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| type-C | 1 | 2 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| type-D | 0 | 1 | 2 | 3 | 2 | 1 | 0 | 0 | 0 | 0 |
| type-E | 0 | 0 | 1 | 2 | 3 | 2 | 1 | 0 | 0 | 0 |
| type-F | 0 | 0 | 0 | 1 | 2 | 3 | 2 | 1 | 0 | 0 |
| type-G | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 2 | 1 | 0 |
| type-H | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 2 | 1 |
| type-I | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 2 |
| type-K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 |

mit Methode meinte er – das geht aus seinen anschließenden Ausführungen hervor – insbesondere die Berücksichtigung seiner Forderung nach dem geschlossenen Fund und eine gute Typologie. Montelius Aussage ist weiterhin gültig. Heute würde ich allerdings spezifizieren: Die Gliederung des Fundmaterials durch eine Typologie muss der spezifischen Fragestellung angemessen sein. Geht es um Chronologie, sollten die Typen im Hinblick auf die Zeit empfindlich sein. Geht es um andere Fragen wie z. B. den sozialen Status, sollte eine Typologie für eben diesen Aspekt optimiert sein. Daher gibt es nicht die eine gute und wahre Gliederung für ein Fundmaterial, sondern jeweils mehrere mögliche geeignete Typologien – jeweils optimiert für eine spezielle Fragestellung. Eine Fibel beispielsweise wird man nach ihrem Stil klassifizieren, wenn es um die Chronologie geht; geht es um eine soziale Fragestellung, könnte ihre Klassifikation nach dem Material (Bronze, Silber, Gold) oder dem Gewicht weitaus zielführender sein; geht es hingegen um das soziale Geschlecht, wird man Fibeln möglicherweise nach ihrer Anzahl und ihrer Trachtlage klassifizieren. Eine CA z. B. mit chronologischer Fragestellung kann nur dann erfolgreich sein, wenn die zu Grunde liegende Typologie angemessen ist. Während des Arbeitens mit einer CA wird man möglicherweise versuchen, die Typologie weiter entsprechend der Fragestellung zu optimieren. Kurz: Wesentliche Teile des Erfolgs einer Studie hängen weniger von den statistischen Berechnungen ab, als vielmehr von der Archäologie und den Arbeiten im Vorfeld der CA, und damit insbesondere von der Typologie.

Mit zu wenig Material wird man ambitionierte Ziele nicht erreichen können. Die Frage, “Wie viel Material braucht man denn?” wird sich kaum allgemeingültig beantworten lassen. Aber die CA kann helfen, sich einer Antwort zu nähern. Wie? Indem man die Stabilität von Ergebnissen beobachtet. Wann immer ein reales Projekt durchgeführt wird, wird man nach der ersten Eingabe Schritt für Schritt versuchen, die Ergebnisse zu optimieren. Dies ist ein wichtiger Teil der meisten Projekte, auch wenn er in den späteren Publikationen leider selten beschrieben wird. Bei einer CA wird man nach Bereinigung der unvermeidlichen Eingabefehler und ersten weiteren Optimierungen irgendwann – so hoffe ich – in einen Zustand geraten, in dem wohlüberlegte weitere Detailverbesserungen das Gesamtergebnis nur mehr unwesentlich verändern. Es gibt keinen Grund für einen Bearbeiter, dann ob der Fruchtlosigkeit seiner Bemühungen enttäuscht zu sein. Vielmehr sollte man sich freuen und Vertrauen in die eigenen Ergebnisse fassen: Das Stadium der Stabilität ist erreicht. Fügt man dann beispielsweise noch neues Material hinzu, z. B. aus einem soeben erst erschienenen Aufsatz, wird es in die Tabelle

eingeordnet, aber die Tabelle insgesamt bleibt in etwa so, wie sie schon vordem war. Wenn eine Analyse dieses Stadium erreicht hat, ist sie stabil, und Ihr Material ist umfangreich genug.

Was tun, wenn man das Material nicht z. B. durch Neufunde erweitern kann, um wie beschrieben die Stabilität einer vorliegenden CA zu testen? Nun, man kann den umgekehrten Weg gehen: nämlich experimentell einzelne Befunde oder Typen aus dem analysierten Datenkörper löschen und schauen, zu welchen Ergebnissen das führt. Wenn der Effekt auf das Gesamtergebnis gering ist, ist das Stadium der Stabilität erreicht und das Material hinreichend umfangreich. Dieses Konzept mag auf den ersten Blick handgestrickt erscheinen, aber das ist ein Irrtum. Im Gegenteil, in der Statistik wird Vergleichbares systematisch durchgeführt; man nennt diese Verfahren Jack-knifing & Bootstrapping (Efron & Tibshirani 1993; Chernick 1999; Good 2013). Jack-Knifing meint, einzelne Fälle (Gräber oder Typen) aus einem Datensatz zu löschen; Bootstrapping meint, dies wiederholt und systematisch zu tun und die Ergebnisse nach jedem Schritt zu beobachten. Man lösche Fall Nr. 1 aus dem Datensatz, führe die Analyse durch und protokolliere das Ergebnis. Anschließend füge man Fall Nr. 1 wieder in den Datensatz ein und lösche statt dessen Fall Nr. 2, führe die Analyse durch und protokolliere das Ergebnis. Und so weiter, bis der komplette Datensatz durchgetauscht ist. Am Ende kann man alle Ergebnisse aus allen Schritten miteinander vergleichen und sehen, ob sie ähnlich zueinander ausfallen, d. h. insgesamt stabil sind. Dabei kann es interessant sein zu beobachten, ob das Herausnehmen einzelner Fälle das Ergebnis mehr als üblich beeinflusst – was dann auch vom archäologischen Standpunkt her zu bewerten wäre. Man könnte daran z. B. schwache Punkte in der Datentabelle identifizieren wie etwa einzelne unscharf definierte Typen, problematische Inventare (z. B. mit Verdacht auf Vermischung). Aber es kann auch sein, dass einzelne Inventare eben einflussreicher sind als üblich, ohne dass dies ein archäologisch-inhaltlicher Fehler wäre.

Das hier Vorgestellte systematisch zu tun, d. h. Grab für Grab und Typ für Typ durchzuspielen, kann ohne Automatisierung Wochen Ihrer kostbaren Zeit verbrauchen! Aber es gibt Wege, diese Prozesse zu automatisieren. Wer in diese Richtung plant, sollte jedoch PAST auf die Seite legen und sich ernsthaft mit **R** auseinander setzen, um entsprechende Skripte zu schreiben, die das Löschen und Wiedereinfügen automatisch durchführen und vor allem den Vergleich der Resultate vornehmen (Good 2013).

Auch wenn man diesen (sehr) aufwändigen systematischen Weg nicht geht, möchte ich dazu ermuntern, mit einem gereiften Datensatz eine kurze Zeit lang in diesem Sinne zu "spielen", d. h. zu experimentieren, um zumindest einen subjektiven Eindruck von der Stabilität oder gegebenenfalls auch von der Instabilität der Ergebnisse zu gewinnen.

### Prüfhypothesen

Doch so schön ein statistisch gutes und stabiles Ergebnis auch sein mag: Letztlich ist vor allem die archäologische Validierung der Ergebnisse entscheidend. Daher benötigt man schon am Beginn eines Projekts Prüfhypothesen, d. h. externe Hypothesen, an denen man die Qualität der Ergebnisse einer CA prüfen kann. Im Falle einer chronologischen Fragestellung könnten

dies z. B. bewährte bisherige Chronologien zum bearbeiteten Material sein, an deren Verfeinerung Sie nun gerade arbeiten, oder eine Chronologie in einem unmittelbar angrenzenden Nachbargebiet. Es könnten aber auch einzelne stratigrafische Beobachtungen sein oder externe Daten, wie sie sich etwa aus $^{14}$C-Daten oder der Dendrochronologie ergeben. Bei meiner Arbeit an der Chronologie der Merowingerzeit am Niederrhein (Siegmund 1998) waren es die vor der CA unternommenen chorologischen (belegungschronologischen) Analysen geeigneter Gräberfelder, die als Prüfhypothese verwendet werden konnten. Man benötigt solche externen Informationen keinesfalls für alle Gräber und Typen einer Tabelle, sondern nur für eine kleine Untermenge. Aber man sollte solche Prüfhypothesen entwickeln, denn man benötigt sie auch während des Arbeitsprozesses mit einer CA, um verschiedene Versuche miteinander vergleichen und bewerten zu können. Ich empfehle sehr, sich gleich zu Beginn einer Analyse systematisch Gedanken über solche Prüfhypothesen zu machen und dies auch in der späteren Publikation offen zu legen.

Wenn man an größeren Tabellen und Streuungsdiagrammen arbeitet, ist es sehr nützlich, vorhandene Prüfhypothesen und externe Informationen schnell sehen zu können. Ich empfehle, die Benennung der Gräber und Typen entsprechend dieses Bedürfnisses vorzunehmen bzw. zu modifizieren, z. B. indem man spezielle Zeichen in die Namen aufnimmt – beispielsweise so, wie es hier in den Datensätzen 6_Langweiler-2_Stehli-1973-p91-fig49 und 7_Schretzheim-beads_Koch-U-1977-table-4 geschehen ist, wo die bereits bestehende Chronologie als erste Ziffer dem Befund- bzw. Grabnamen vorangestellt ist. Auf diese Weise kann man unmittelbar erkennen, wie die von der aktuellen CA gewonnenen Ergebnisse mit den älteren Studien zusammenpassen.

PAST enthält eine Option, die hilfreich sein kann, mit Prüfhypothesen zu operieren. Man kann in einem Datensatz eine "Gruppierungsvariable" definieren. Sie wird bei der Berechnung einer CA mathematisch nicht berücksichtigt, kann aber in das Bild der Streuungsdiagramme eingespielt werden. Wer neugierig ist, möge es selbst versuchen; hier sei nur der Weg skizziert: Eine neue Spalte hinzufügen; bei *Show* die *Column attributes* aufrufen; dort unter *Name* die Spalte angemessen benennen und bei *Type* (Voreinstellung "–") aus dem Menü *Group* auswählen; vor dem Variablennamen sollte danach ein blaues "G" erscheinen. Dann die CA für die Daten inkl. dieser Gruppierungsvariable berechnen. Im Fenster *Correspondence analysis* in der rechten Leiste unten das Häkchen bei *Group labels* einschalten: Die Gruppierungsvariable wird in roter Schrift eingeblendet, entsprechend des Schwerpunktes, den die Vertreter dieser Gruppe bei der CA gemeinsam gewonnen haben.

### 12.3 Welche Eingriffe sind erlaubt, was sollte man nicht tun? Einige praktische Hinweise

Keine Datentabelle und keine CA ist gleich beim ersten Ansatz fertig. In den meisten Fällen fußt das später publizierte Ergebnis auf einem längeren Arbeitsprozess mit vielen Experimenten der Art "Versuch und Irrtum" (*trial & error*). Was kann und darf man tun, und was würde gegen die Regeln guter wissenschaftlicher Praxis verstoßen? Es ist nicht korrekt, nach Durchführung einer CA die anhand der CA geordnete Tabelle manuell nachzuordnen,

um einzelne Typen oder Inventare dahin zu schieben, wohin sie aus Sicht des Bearbeiters "eigentlich" gehören. Es ist nicht korrekt, mit dem gleichen Ziel einzelne "störende" Objekte aus einzelnen Inventaren zu löschen. Aber man kann Typen in Gänze wieder aus der Analyse herausnehmen und man kann auch ganze Inventare wieder aus einer Analyse herausnehmen. Gut wäre es, wenn man dafür zuvor Regeln umrissen hat, die man auch vertreten kann und publiziert. So hat man z. B. manchmal bei Fundensembles den Verdacht, sie könnten durch Verwechslungen oder Vermischungen in einem Museumsmagazin beeinträchtigt sein, wie es beispielsweise bei Perlenketten schon vorgekommen sein soll. In solchen Fällen kann eine insgesamt bereits einigermassen stabile CA Hinweise darauf geben, ob ein bereits unter solchem Verdacht stehendes Inventar tatsächlich vermischt ist; wird der Verdacht dann via CA erhärtet, ist es richtig, es aus der weiteren Analyse auszuschließen. Ähnlich kann bei z. B. Siedlungsgruben bereits auf der Grabung der Verdacht entstanden sein, dass man möglicherweise zwei Befunde nicht hat trennen können. Auch hier wäre es berechtigt, einen solchen Befund nach einem via CA bestätigten Vermischungsverdacht aus der weiteren Analyse auszuschließen. Über "Langläufer" hatten wir schon im Kap. 9.1 gesprochen: Typen, die zwar wohl definiert, aber in Bezug auf die Fragestellung zu unsensibel sind. Solche Typen können ohne weiteres aus einer Analyse ausgeschlossen werden. Hilfreich ist es, wenn man hierfür vorab Kriterien definiert hat, damit das gesamte Material diesbezüglich gleich behandelt wird.

Es ist manchmal schwierig und langwierig, via Versuch und Irrtum den geeigneten Datensatz für die endgültigen Ergebnisse herauszudestillieren. Sind die Typen sehr spezifisch und chronologisch eng umrissen, könnten sich zu wenige Fundkombinationen ergeben. Integriert man zu viele unspezifische Typen in einen Datensatz, mehrt man zwar die Anzahl der Fundkombinationen, verschleiert aber möglicherweise die Feinchronologie. Es gibt keine festen Regeln für das Vorgehen. Hilfreich sind: explizite Prüfhypothesen, gezielte Versuche und ausformulierte Wege und Kriterien für das Vorgehen.

Ein weiteres Caveat ist angebracht: Man kann recht viel Zeit mit den Versuchen einer Optimierung verbringen. Es ist lohnend, beizeiten zufrieden zu sein und die Arbeit an der CA zu beenden. Möglicherweise hilft es, bereits vor Beginn einer Analyse Kriterien festzulegen, wann man mit dem weiteren Optimieren aufhören möchte. Die oben in Kap. 12.2 behandelten Themen "Erreichen der Stabilität" und "Prüfhypothese" können dabei hilfreich sein.

Was kann man tun, wenn Teile der Tabelle zu wenige Fundkombinationen aufweisen, also zu gering miteinander verbunden sind (Kap. 9.3)? Man kann versuchen, zusätzlich externes Material in die Tabelle aufzunehmen, beispielsweise publiziertes Fundmaterial von nahegelegenen Fundorten. Man kann die eigene Typologie kritisch überdenken: Vielleicht ist sie zu empfindlich und führt zu allzu feingliedrigen und daher kleinen Gruppen? Bei komplexen Fundgattungen kann man erwägen, einzelne Typen in Merkmalsgruppen aufzulösen und dann statt der Typen mit den Merkmalen weiter zu arbeiten. So kann man beispielsweise im Frühmittelalter statt kompletter Typen von Gürtelschnallen diese auch einmal nach ihrer Form klassifizieren und einmal nach ihrer Verzierung und anschließend mit den Typen der Gürtelformen und jenen der Gürtelverzierung eine Analyse durchführen. Solch ein Ansatz kann helfen, dünn besetzte Schwachstellen in einer Tabelle zu überbrücken.

Manchmal ist es hilfreich, die Regel "jedes Inventar enthält mindestens zwei Typen und jeder Typ ist in mindestens zwei Befunden vertreten" (Kap. 9.3) zu verschärfen und diese Schranke auf drei oder vier hochzusetzen (so z. B. bei Siegmund 2013). Gerade bei der Analyse von Siedlungsmaterial kann dies helfen, Singularitäten aus dem Datensatz zu entfernen.

Andererseits kann es sinnvoll sein, allzu umfangreiche Komplexe aus einer Analyse auszuschließen. Wenn beispielsweise in einem großen Komplex von Siedlungsgruben üblicherweise z. B. drei bis zehn typisierte Funde pro Inventar vorkommen, dürfte ein Inventar mit z. B. 100 typisierten Funden zu Recht unter dem Verdacht stehen, ein ungewöhnlich lange offen liegender "Materialsammler" zu sein, dessen Inventar chronologisch unspezifisch ist. Man kann diese Frage nach dem üblichen Umfang der bearbeiteten Inventare bereits vor Durchführung der CA angehen und klären, wo eine solche Maximalschranke begründet gesetzt werden sollte.

Bei einem realen Projekt, bei dem man erwartungsgemäß eine gewisse Zeit lang intensiv an einer Tabelle arbeiten wird, stellt sich am Beginn oft die Frage, wie man beginnen soll: mit dem Gesamtmaterial, um dann via Versuch und Irrtum sukzessive einzelne weniger taugliche Typen und Befunde wieder herauszunehmen – oder umgekehrt mit einem sicheren Kern, den man anschließend in sukzessiven Versuchen erweitert. Eine eindeutige Antwort auf diese Frage gibt es nicht. In der beratenden Beobachtung verschiedener Projekte meine ich wahrgenommen zu haben, dass es für sich unsicher fühlende Anfänger leichter sein könnte, mit einem sicheren Kern zu starten und diesen allmählich zu erweitern, während es für erfahrene Bearbeiter effizienter ist, mit dem Gesamtkomplex zu beginnen, um diesen dann so weit als nötig zu lichten. Eine gute Selbstprüfung kann es sein, sein Vorgehen vorab schriftlich zu fixieren mit dem Ziel, diesen Arbeitsplan oder auch Versuchsaufbau später zum Teil der Publikation zu machen. Denn beim Schreiben zeigt sich schnell, inwieweit die ersten Ideen zum Vorgehen auch Dritten gegenüber als gut vertretbar erscheinen – wie etwa die Frage, was warum als sicherer Kern für den Start des Verfahrens herangezogen wurde. Dabei geht es nicht um einzelne Typen oder Befunde, sondern um die explizite Darlegung des Vorgehens insgesamt und der Kriterien und Regeln, denen man dabei folgen möchte. Kann man die Auswahl eines Kernmaterials gut begründen? Oder kann man alternativ vorab gut darlegen, nach welchen Kriterien man Typen oder Befunde nach einem Start mit dem Gesamtmaterial wieder aus der Analyse ausschließen möchte? Wer versucht, diese Pläne vorab niederzuschreiben, findet erfahrungsgemäß schnell den für sich richtigen Weg, den er beschreiten möchte.

### 12.4 Der Eckeffekt, und wie man damit umgehen kann

Erfahrungsgemäß ist die Ordnung einer Tabelle an ihren Rändern nicht optimal. Bisweilen steht gerade am Anfang und am Ende einer chronologischen Sequenz nur vergleichsweise wenig Material zur Verfügung, so dass die Belegdichte dort erheblich geringer ist als im Mittelbereich einer Tabelle. Doch selbst dann, wenn dieser Effekt weniger gravierend ist, fällt die Ordnung an den Rändern bisweilen unbefriedigend aus. Dies dürfte auch damit zusammenhängen, dass

einige der "frühen" resp. "späten" Typen in der historischen Wirklichkeit ebenfalls mit noch älterem resp. noch jüngerem Material kombiniert waren, genau diese Fundkombinationen aber nicht mehr in der vorliegenden Tabelle vertreten sind – eben weil sie jenseits ihrer Ränder liegen. Für das statistische Herausfinden ihrer historisch wahren chronologischen Anordnung mit Hilfe der CA fehlen diese Komplexe und Typen jenseits der Ränder der Tabelle, so dass die Positionierung der in der Tabelle enthaltenen randlichen Komplexe und Typen nur durch jene Typen und Komplexe gesteuert werden, die innerhalb der Tabelle liegen, d. h. die letztlich in Richtung auf die Tabellenmitte "ziehen". Was kann man tun? In vielen Fällen ist es sinnvoll, die Tatsache einer tendenziell schwächeren Ordnung in den Randbereichen einer Tabelle schlicht zu akzeptieren. Doch was kann man unternehmen, wenn genau diese Bereiche für die verfolgte historische Fragestellung besonders wichtig sind? Einfache Antwort: Die Ränder nach außen verschieben. Durch das gezielte Hinzufügen weiteren Materials unmittelbar jenseits der aktuellen Ränder, z. B. durch das Hineinnehmen weiterer, noch älterer resp. noch jüngerer Komplexe aus benachbarten Siedlungen oder Gräberfeldern, kann man versuchen, den im Interesse liegenden Bereich vom Rand mehr in die wohlgeordnete Zone der Tabelle zu verlagern, so dass der Eckeffekt nun eher auf das neu hinzugefügte Material wirkt. Das geht nicht immer, aber durch den Hinweis hier ist zumindest die Frage angestoßen, darüber nachzudenken.

## 12.5 Über Detrending, Gewichten und Kanonische Korrespondenzanalyse

Der bisherige Text galt der weithin üblichen normalen CA, so wie sie in vielen archäologischen Projekten zu guten Ergebnissen geführt hat. Bisweilen werden spezielle Eingriffe in die gewöhnliche CA vorgenommen oder Varianten derselben resp. verwandte Verfahren angewendet. Bei den üblichen Fragestellungen und Daten sind solche speziellen Lösungen nicht notwendig, und ich rate sehr dazu, nur zögernd in deren Anwendung einzutreten und dann sehr gute Gründe dafür zu haben. Um aber Lesern, die eventuell bereits interessant klingende Stichworte aufgelesen haben oder die vorliegende Publikationen besser verstehen wollen, Orientierung zu geben, werden hier einige häufiger verwendete Schlagworte und Verfahren kurz erläutert. Nur in Spezialfällen ist deren Anwendung tatsächlich gewinnbringend.

### Detrending

Die wesentliche Bedeutung dieses Begriffs wurde bereits erläutert (Kap. 8.5), das Verfahren nennt man *Detrended Correspondence Analysis* (DCA), und es ist eine Variante der CA. Beim Detrending wird nach Durchführung einer CA die Parabel – eine quadratische Funktion – aus den Achsen 1 und 2 herausgerechnet. Ziel ist es, sowohl auf der Strecke von Achse 1 als auch auf Achse 2 zu genauer bemessenen Abständen zwischen den Zeilen- und Spaltenwerten zu gelangen. Die Anwendung einer DCA ist dann sinnvoll, wenn die Eigenwerte der CA tatsächlich als lineare Skala benutzt werden sollen, etwa zum Schätzen der kalendarischen Zeit oder ähnlichem. Ein anderer Grund für die Anwendung einer DCA ist das Bedürfnis, besser mit Achse 2 arbeiten zu können, die – zumindest in der Theorie – nach einem Detrending besser

interpretierbar sein sollte. Immer wenn Ziele der skizzierten Art ins Auge gefasst werden, kann eine DCA sinnvoll sein. Jedoch bleibt aus meiner Sicht die CA das Standardverfahren der Wahl.

### Gewichtungen

Archäologen fragen erfahrungsgemäß gerne nach der Option von Gewichtungen, um ihre archäologische Erfahrung mit dem Fundgut in eine CA einzubringen. Denn "schließlich weiß man ja", dass bestimmte Gattungen und Typen für eine Chronologie wichtiger sind als andere und möchte dieses Wissen via Gewichtung in die CA einbetten. Der Wunsch erscheint nachvollziehbar und aus statistischer Sicht spricht wenig gegen seine Umsetzung. Die Softwarelösungen WinBASP und CAPCA beispielsweise bieten komfortable Funktionen für solche Gewichtungen. Aber… wirklich begeistert ist der Verfasser vom Gewichten nicht. Jedenfalls sollte das Gewichten von Typen sehr eindeutige und einfache Regeln verfolgen, und es sollten jeweils gute Gründe bestehen, in dieser Weise in eine CA einzugreifen. Solche Eingriffe in die CA müssen in der späteren Publikation benannt werden und die verfolgten Regeln sollten transparent offen gelegt sein. Ein guter Grund wäre z. B. das in Kap. 12.3 erwogene Aufteilen einzelner Objekte (Typen) zu zwei Merkmalsgruppen (einmal Form, einmal Verzierung), womit ein Objekt eine quasi doppelte Präsenz im Datensatz gewönne. Wem dies nicht richtig erscheint, der könnte die beiden Merkmalsgruppen jeweils mit 0.5 gewichten und dadurch deren doppelte Präsenz zurücknehmen. Ein anderes Beispiel ergibt sich aus dem unterschiedlichen Merkmalsreichtum von Objekten. Verzierte Perlen und verzierte Keramik könnte man als prägnantere Aussage aus der Vergangenheit wahrnehmen, einfache unverzierte Perlen oder unverzierte Keramik als unkonkretere Typen, die man nicht mit elaborierten Objekten gleichstellen möchte. Auch hier könnte eine Gewichtung eingebracht werden. Meine persönlichen Erfahrungen mit diesem Thema sind: Gewichten kann sinnvoll sein, aber es führt dazu, dass Tabellen tendenziell weniger transparent lesbar werden, und der Effekt auf die Ergebnisse ist zumeist geringer als zunächst erwartet oder auch erhofft. Mein Rat: halten Sie die Dinge so einfach wie möglich und Gewichten Sie nur in Ausnahmefällen.

Viele Anwender übersehen, dass die CA bereits verfahrensbedingt Gewichtungen vornimmt. Die im Hintergrund jeder CA stehende Chi-Quadrat-Metrik führt dazu, dass ein einzelner Vertreter eines seltenen Typs mehr Einfluss auf das Ergebnis hat als ein einzelner Vertreter eines häufigen Typs, und ebenso ein einzelnes Objekt in einem Inventar mit insgesamt zwei Objekten mehr Bedeutung hat als ein einzelnes Objekt in einem umfangreichen Inventar. Diese Eigenschaft der CA wurde beispielsweise in der Ökologie kritisiert und deshalb wurden andere Maße für die Ähnlichkeit resp. Entfernung der Typen und Befunde vorgeschlagen, welche für ökologische Fragestellungen besser geeignet sein sollen (Legendre & Gallagher 2001). In der Archäologie jedoch erscheint mir gerade diese verfahrensimmanente Gewichtung der CA unseren Vorstellungen und Bedürfnissen angemessen: Merkmalsreiche, scharf definierte Typen werden tendenziell selten sein, aber für eine Chronologie sind sie besonders nützlich, während mehrmalsarme Typen möglicherweise auch häufiger und chronologisch weniger sensibel sind.

Insofern passt die einer CA innewohnende Metrik bestens zu den Vorstellungen von Archäologen über die Bedeutung (Gewichtung) ihrer Typen und Gräber.

### Kanonische Korrespondenzanalyse

Die Kanonische Korrespondenzanalyse (CCA) unterscheidet sich deutlich von der CA. Ihr Ziel ist es, eine gegebene Tabelle unter Annahme des unimodalen Modells zu ordnen, aber zunächst so, dass die Ordnung entlang einer (oder mehrerer) vorgegebenen "kanonischen" Variable optimal nachvollzogen wird. Erst nach Erreichen dieser ersten, "kanonischen" Ordnung entlang der vorgegebenen Variablen werden weitere, "freie" Achsen extrahiert, die den Achsen einer CA ähnlich sind. Sofern also ein unimodales Modell angenommen werden kann und für viele Fälle eine geeignete kanonische Variable vorliegt, kann die Anwendung einer CCA sinnvoll sein. Beispiele? Frühmittelalterliche Grabinventare zeigen einen starken Unterschied nach dem sozialen Geschlecht der Bestatteten, er ist stärker als die zeitbedingten Unterschiede. Üblicherweise wird daher für die Frauen- und Männergräber getrennt je eine CA berechnet – da die Nicht-Kombination von Objekten eben auf großen Zeitunterschieden zwischen den Inventaren beruhen kann, oder eben auf dem Geschlechtsunterschied zwischen zeitgleichen Inventaren, also mehrdeutig ist. In dieser Lage könnte man alternativ zum gängigen Vorgehen erwägen, einen gemeinsamen Datensatz aller Geschlechter per CCA zu analysieren, in dem das soziale Geschlecht als kanonische Variable gesetzt ist, damit dann als erste freie Achse die beiden Geschlechtern gemeinsame Dimension Zeit gewonnen wird. Eine feine Idee. Die diesbezüglichen Versuche des Verfasser sind indes gescheitert, die gewonnene Ordnung war nicht hinreichend gut und den beiden geschlechtsspezifisch getrennt gerechneten CA deutlich unterlegen.

Ein weiterer und interessanter Anwendungsfall für eine CCA sind Pollenspektren aus Bohrprofilen resp. langen Stratigrafien. Hier kann man die Probentiefe als kanonische Achse setzen und gewinnt eine Ordnung der Pollenspektren entlang der Stratigrafie. Einige weitere Anwendungsbeispiele finden sich z. B. bei Müller & Zimmermann (1997). Insgesamt ist die CCA kein Standardverfahren in der Archäologie, kann aber in wohlüberlegten Spezialfällen eine nützliche Alternative zur CA sein.

### Redundanzanalyse

Nachdem wir zuvor die CCA verstanden haben, ist die Redundanzanalyse (RDA) schnell erklärt: RDA ist CCA bei linearem Modell (Jongman, ter Braak & van Tongeren 1995). Bei einer RDA wird ebenfalls zunächst versucht, eine Datentabelle entlang einer (oder mehrerer) kanonischen Variablen optimal zu ordnen, um anschließend weitere freie Achsen zu extrahieren. Anders als die CCA geht eine RDA jedoch vom Vorliegen eines linearen Modells aus.

Der Autor hat eine RDA in zwei unterschiedlichen Situationen auf archäologisches Material angewendet. Einmal bei zu ordnenden Ensembles von Steinartefakten, bei denen ich entlang der kanonischen Achsen ein lineares Verhalten erwartete (Siegmund 1991), wobei als Ordnung

zunächst weniger die Dimension Zeit, sondern mehr funktionale Faktoren und Umweltbedingungen erwartet waren. In einem weiteren Fall mit chronologischer Fragestellung auf Funde aus Schichten einer recht kurzen stratigraphischen Sequenz (Siegmund 1994); hier war zwar generell ein unimodales Verhalten zu erwarten, wovon angesichts der Kürze der in der Schichtenfolge enthaltenen Zeit jedoch nur das Vorliegen einer Hälfte der jeweiligen Glockenkurve vermutet wurde, und dazu wiederum passt mehr das lineare Modell der RDA. So darf die RDA insgesamt für die Archäologie als eher exotisches Verfahren bewertet werden, das nur in seltenen Ausnahmefällen greift.

### Lineares Modell: Hauptkomponentenanalyse (PCA)

Die Hauptkomponentenanalyse (PCA) ist die einfachste Variante einer Faktorenanalyse und das Pendent zu einer CA, jedoch bei Annahme eines linearen Modells. Ähnlich wie bei einer CA werden mehrere voneinander unabhängige Achsen – hier Komponenten genannt – aus dem Datenkörper extrahiert, welche in absteigender Bedeutung die grundlegenden Strukturen hinter den beobachteten Daten aufdecken sollen. Wer an Stelle langer theoretischer Ausführungen schnell experimentell nachvollziehen möchte, was geschieht, wenn in der Frage unimodales Modell oder lineares Modell (d. h. CA oder PCA) eine unangemessene Entscheidung getroffen wird, kann dies dank PAST mit Hilfe einer PCA recht schnell ausprobieren. Unser Datensatz 1b_ideal-matrix-unordered ist nach dem unimodalen Modell konstruiert und kann mit einer CA bestens geordnet werden (**Kap. 7**). Führen wir eine PCA mit diesem Datensatz durch und prüfen die resultierende Ordnung. Man starte PAST, lade den Datensatz 1b_ideal-matrix-unordered, markiere die ganze Tabelle als ausgewählt, » *Multivariate*, » *Ordination*, » *Principal component (PCA)*. Im neu aufklappenden Fenster *Principal component analysis* klicke man auf den Reiter *Scatter plot* und studiere das Bild. Entlang der dominanten Achse (waagerecht, *Component 1*) lautet die Ordnung der Gräber von links nach rechts: 3, 2, 4, 1, 5, 6, 10, 7, 9, 8. Es liegt also im Vergleich zu unseren Erwartungen eine erhebliche Unordnung vor, was im übrigen auch für die Typen gilt und auch dann, wenn wir die zweitwichtigste Achse (senkrecht, *Component 2*) studieren. Bildlich erklärt: wenn man die Daten einer Glockenkurve nach dem linearen Modell untersucht, wird diese im Grunde in der Mitte gefaltet, beide Extreme (Anfang und Ende) bilden gemeinsam das eine Ende der PCA-Achsen, das Maximum der Glockenkurve das andere Ende: Chronologie-Chaos statt Ordnung. Aber auch andersherum wird ein Schuh draus: Daten, die tatsächlich dem linearen Modell folgen, sollten nicht mit einer CA untersucht werden. Wer sich nun vertiefend mit der Hauptkomponentenanalyse (PCA) resp. der Familie der Faktorenanalysen beschäftigen möchte, sei auf gute Einführungen in die multivariate Statistik verwiesen (z. B. Hartung & Elpelt 2007; Hair, Black, Babin & Anderson 2010).

## 13 Übernehmen der Ergebnisse einer vorliegenden CA

In manchen Fällen möchte man die Ergebnisse einer anderweitig existierenden CA nur auf das eigene Material übertragen. Ein üblicher Grund dafür ist beispielsweise der Eindruck, dass

das eigene Material für eine eigenständige Analyse in zu geringer Zahl vorliegt und in dem hinzugezogenen Referenzwerk die CA wiederum gut begründet und durchgeführt ist. Wie geht man in solchen Fällen vor? Es gibt drei unterschiedliche gute Optionen:

(1) Man hält es einfach und geht ohne Statistik vor. Man liest die Referenzstudie, analysiert die Phasenbildung und die Zuordnung der Typen zu diesen Phasen. Dann überträgt man sein Material ohne weitere Statistik in diese Phasen. Das ist der übliche Weg, und er ist oft auch angemessen.

(2) Man übernimmt die Datentabelle aus dem Referenzwerk, fügt seine eigenen Daten hinzu und rechnet die CA neu. Mit diesem Ansatz sollten die eigenen Typen und Inventare perfekt in die Ordnung des Referenzwerkes eingefügt werden können. Aber durch die neuen Daten wird die Ordnung des Referenzwerkes leicht oder stärker verändert werden, was im wesentlichen von der Menge der neu hinzugefügten Daten abhängig sein wird. Dies kann z. B. Auswirkungen auf die Phasenbildung und -zuordnung haben, die dann zwischen beiden Analysen nicht mehr ganz passt. Wer das unbedingt vermeiden will, wähle Option (3).

(3) Man wendet die (Eigen-)Werte (*scores*) aus dem Referenzwerk auf das eigene Material an. Die Position jedes Typs und jedes Grabes im Achsenraum des Referenzwerkes lässt sich anhand der (Eigen-)Werte exakt berechnen, ohne dabei die Ordnung des Referenzwerkes zu tangieren. Die neuen Typen und Inventare sind dann in der Sprache der Korrespondenzanalyse *supplementary points* (Greenacre 2007, 89-96). Wie das geht? Für die Positionsbestimmung eines Inventars/Grabes nehme man den Eigenwert jedes Typs in dem Referenzwerk, multipliziere diesen Wert mit der Typhäufigkeit in dem aktuell zuzuordnenden Inventar, addiere diese Werte und dividiere die resultierende Summe durch die Summe der typisierten Objekte in diesem Inventar. Das Ergebnis entspricht dem Eigenwert des Grabes entlang Achse 1 im Referenzwerk. Klingt aufwendig, ist aber mit Hilfe einer Tabellenkalkulation recht schnell umzusetzen. Wem diese Skizze zu allgemein war, lese die Details bei Greenacre (2007, 89-96) nach.

Wie die Option (2) zeigt, ist es wertvoll, wenn die originale Datentabelle in einer bequem elektronisch lesbaren Form zur Verfügung steht. Option (3) zeigt, dass die Eigenwerte der Typen und Befunde/Gräber (*scores*; früher auch "Schwerpunkte" genannt) stets publiziert werden sollten, damit Andere sie wie beschrieben weiterverwenden können.

## 14 Schlussbemerkung zum ersten Teil

Auf den ersten Blick scheinen die Theorie und die Durchführung einer CA kompliziert zu sein. Ich hoffe, dass dieser Eindruck nach dem Durcharbeiten dieses Leitfadens nicht mehr fortbesteht und Sie sich in der Lage fühlen, z. B. mit Hilfe von PAST eine CA auch selbständig durchzuführen. Der Kern einer solchen Arbeit sollte stets die Archäologie sein, hier liegt der entscheidende Schlüssel für eine gute Studie. Für den Anfang ist es nützlich, sich

ein publiziertes, dem eigenen archäologischen Problem ähnliches Beispiel einer CA-gestützten Analyse zu suchen und als Vorbild zu benutzen, das man nacharbeitet, ähnlich wie man ein Kochbuch heranzieht. Eine bessere Hilfe ist ein erfahrener Kollege, den man von Zeit zu Zeit um eine Diskussion der Zwischenergebnisse bitten kann. Entscheidend ist aber, jetzt einfach den Mut zu finden und mit einem eigenen Projekt loszulegen. Denn noch so optimale Übungsdaten sind nie so motivierend und weiterführend wie ein eigenes Problem, an dem man sich tiefer in das Thema CA einfuchst.

## 15 Anregung für eine weiterführende Lektüre

Eine kurze Einführung in das Thema Seriation (Anwesenheits-/Abwesenheits-Seriation und Häufigkeitsseriation) und CA in deutscher Sprache findet sich bei Ihm (1983). Das Besondere dieses Aufsatzes sind kleine Beispiele, an denen Ihm Schritt für Schritt aufzeigt, wie diese Statistiken wirklich gerechnet werden. Nach Durcharbeiten dieses Aufsatzes hat man verstanden, was bei einer CA im Hintergrund gerechnet wird. Eine deutlich mathematischere, englischsprachige Einführung in das Thema bietet Ihm & Groenewould (1984). Als gut geschriebene, kurze Einführung in die Geschichte der Methodenentwicklung empfehle ich Ihm (2005) und ergänzend - weil aus anderer Perspektive - den entsprechenden Abschnitt bei de Leeuw, J. (2013).

Eine gute Einführung in den aktuellen Stand der Methodologie findet sich in dem Buch von Müller & Zimmermann (1997). Das Werk bietet zudem verschiedene Anwendungsbeispiele der CA auf unterschiedliche archäologische Materialien und Fragestellungen.

Zwei Beiträge über frühmittelalterliche Perlen seien hier erwähnt, weil dieses Thema mit einer höheren Bedeutung des Aspekts der Häufigkeit von Typen einhergeht als es beispielsweise in Grabinventaren der Fall ist (Siegmund 1995; Sasse & Theune 1996). Daher sind Studien zu Perlen und Perlenketten besser mit denen zu Keramikensembles in Siedlungskomplexen vergleichbar.

Manchmal ist es anregend, einen Blick auf die US-amerikanische Diskussion zum Thema Seriation zu werfen, die sich nicht eng am europäischen Diskurs orientiert, sondern recht eigenständig verläuft. Die Debatte dort ist tief geprägt vom Werk von J. A. Ford (1962). Drei neuere Aufsätze seien hier aufgeführt, die einen schnellen Einstieg in die dortige Debatte ermöglichen: O'Brien et al. (2000), Smith et al. (2007) und Lipo et al. (2015).

## 16 Anregungen für weitere Trainingsfälle zum Ausbau der praktischen Erfahrungen

Möglicherweise möchten Sie Ihre praktischen Erfahrungen noch etwas weiter an guten Übungsbeispielen ausbauen, bevor Sie ein Projekt mit eigenem Material beginnen. Dazu einige Anregungen zum Training an echtem archäologischem Material:

(1) Analysieren Sie die klassische Studie von Oscar Montelius (1885) über die Chronologie der nordischen Bronzezeit, in der er die Grundlagen für das bis heute verwendete Chronologiesystem legte. Sein Buch umfasst sorgfältig geführte Tabellen der Grabfunde und Typen, die recht einfach in eine Computertabelle übertragbar sind (Montelius 1885, 270-311); die Publikation steht heute auch als Scan via Internet kostenfrei zur Verfügung. Inzwischen gibt es auch eine Übersetzung in die englische Sprache (Montelius 1996), der jedoch leider die wichtigen Listen fehlen. Doch mit beiden Ausgaben zusammengenommen ist alles Nötige auch ohne Kenntnis der schwedischen Sprache hinreichend verständlich.

(2) Schöne Beispiele für konventionell handsortierte "Kombinationstabellen" mit chronologischer Fragestellung enthält die Studie von J. Giesler (1981, insbes. Tab. 30, 34, 40, 45 und 52). Es lohnt, diese nicht allzu großen Tabellen mit Hilfe einer CA nachzuvollziehen, die Ergebnisse mit der Sortierung von Giesler zu vergleichen und darüber nachzudenken, wie man diese Ordnungen mit den dort ebenfalls untersuchten Münzchronologien und den Belegungschronologien verknüpfen kann.

(3) Eine dritte schöne Fallstudie behandelt die Entwicklung von Form und Verzierung von Römern, also frühneuzeitlichen Weingläsern, die in datierten niederländischen Stilleben des 17. Jahrhunderts dargestellt sind. Die Originalarbeit stammt von Brongers & Wijnman (1968). Deren Daten wurden von Goldmann (1972, 29-33) sowie von Eggert et al. (1980) als Testdatensatz für ihre methodische Debatte über die Seriation verwendet. Hier liegt die besondere Herausforderung weniger im Rechnen einer CA, sondern in der Frage, wie man die vorliegenden Informationen adäquat formalisiert und bestmöglich nutzt.

## 17 Ziel erreicht

Wir sind am Ende des ersten Teils dieses Praxisleitfadens angelangt. Möglicherwiese haben Sie nun doch etwas länger als die eingangs genannten etwa acht Arbeitsstunden darauf verwendet. Aber Sie verfügen jetzt über alle notwendigen Kenntnisse und Fertigkeiten, um Ihr eigenes Projekt zu starten. Sie sind kein Anfänger mehr, sondern eine Archäologin mit wachsenden Erfahrungen in der Anwendung der Korrespondenzanalyse. Das nächste eigene Projekt macht Sie zur erfahrenen Expertin.

## 18 PAST: Das semi-automatische Sortieren großer Tabellen

Leider enthält PAST (noch?) keine Funktion zur automatischen Neusortierung einer Eingabetabelle nach den Ergebnissen einer CA. Obwohl PAST über alle Instrumente verfügt, dies immerhin semi-automatisch zu tun, ist mir die Umsetzung trotz mehrfacher Bemühungen nicht gelungen. Sollte ich etwas übersehen haben, würde ich mich über eine Rückmeldung sehr freuen. Daher skizziere ich nachfolgend einen Weg, solche Sortierungen außerhalb von

PAST mit Hilfe der gängigen Tabellenkalkulationsprogramme durchzuführen. Anders als bisher beschreibe ich den Weg jedoch nicht detailliert Klick für Klick und Schalter für Schalter, sondern skizziere den Lösungsweg nur grundsätzlich. Nach einem ersten Durchlauf dürfte das Prozedere recht schnell zur Routine werden und die erwünschte Neuordnung jeweils innerhalb weniger Minuten erreicht sein. Aus lizenzrechtlichen Gründen entwickle ich den Lösungsweg hier anhand des Open-Source-Programmes LibreOffice Calc, versichere aber, dass mir Gleiches auf ähnlichem Weg z. B. auch mit MS-Excel gelungen ist.

Ziel ist es also, die Ursprungstabelle und die Ergebnisse (d. h. die *scores*) der CA zusammen zu bringen und die Tabelle dann einmal nach den Zeilen und einmal nach den Spalten zu sortieren. Dazu sind folgende Schritte nötig:

(1) Export der Ursprungstabelle aus PAST in die Tabellenkalkulation. Am sichersten geht dies über das Format xls. Also *File*, » *Save as…* und unten den Dateityp *.xls auswählen.

(2) Import der Ursprungstabelle nach Calc: Datei, » Datei öffnen, … Die von PAST übernommene zusätzliche Kopfzeile vor der Zeile mit den Typ-Namen und die führenden Spalten vor jener mit den Grab-Namen löschen.

(3) Aus PAST, und zwar aus dem Fenster *Correspondence analysis*, die *Row scores* kopieren. Leider kann man dort keine einzelne Spalte kopieren, sondern nur die ganze Tabelle, und zwar mit Hilfe der Schaltfläche ganz unten im Fenster, *Copy*.

(4) Den Inhalt des Zwischenspeichers nun in Calc einfügen, am besten rechts oben neben der schon bestehenden Tabelle. Mir war der Weg über "unformatiert einfügen" sympathisch und erfolgreich. Prüfen Sie, ob bei der Ausgangstabelle und den nun eingefügten Scores die Zeilen zueinander passen. Eventuell sitzt die neu eingefügte Teil-Tabelle mit den Scores z. B. eine Zeile zu hoch oder zu tief. Bereinigen Sie dies ggf. mit einem erneuten Copy & Paste. Löschen Sie alle Scores und Spalten, die nicht gebraucht werden; benötigt wird nur die Spalte *axis 1*.

(5) Prüfen Sie vorsichtshalber, ob die eingefügten Zahlen auch Zahlen im technischen Sinne sind: Spalte oder Ausschnitt markieren, Zellen formatieren und ggf. z. B. statt Text auf Zahl einstellen.

(6) Tabelle nach der Spalte *axis 1* sortieren. (» *Daten*, » *Sortieren*, dort bei Sortierschlüssel die richtige Spalte einstellen", OK). Erster Schritt erfolgreich erledigt.

(7) Da in PAST die Scores der Typen ebenfalls in Zeilen vorliegen, sind sie jetzt nicht per einfachem Copy & Paste in die neue Calc-Tabelle übertragbar. Daher muss eine der beiden Tabellen zunächst einmal um 90 Grad gedreht werden, d. h. aus Spalten sollen Zeilen und aus Zeilen Spalten werden. Wie?

(8) In PAST bei *Correspondence analysis* den Reiter *Column scores* aktivieren und den Inhalt mit Copy (unten im Fenster) in die Zwischenablage schreiben.

(9) Unter Calc eine neue (Hilfs-) Tabelle anlegen und den Inhalt des Zwischenspeichers einfügen. Überflüssiges löschen, so dass nur noch zwei Spalten übrig bleiben: die Liste der Typen und die Scores der Achse 1.

(10) Diese Hilfstabelle nun "transponieren" (*transpose*) – so lautet der Fachbegriff, wenn Sie z. B. in anderen Programmen via Hilfefunktion weiter kommen möchten. In Calc geht dies recht einfach: Die verbliebene Tabelle markieren und ausschneiden; für das Wiedereinfügen die rechte Maustaste klicken, Inhalte einfügen, dort Häkchen setzen bei: Alles einfügen und unten bei Transponieren, die OK-Schaltfläche bestätigen. Fertig. Wenn die resultierende, bereits transponierte Tabelle jetzt zu viele Spalten / Zeilen beinhaltet: Überflüssiges löschen. Erhalten bleiben sollten nur zwei Zeilen: die Zeile mit den Typ-Namen und die Zeile mit den Scores. Wenn Sie skeptisch oder vorsichtig sind: Man kann jederzeit mit Blick auf die Scores unter PAST prüfen, ob alles richtig verlaufen ist.

(11) Die beiden Zeilen der Hilfstabelle markieren, kopieren und bei der Datentabelle unter deren unterste Zeile einfügen. Achtung: Dieses Mal das Häkchen bei Transponieren wieder entfernen! Prüfen, ob die Typenliste der ursprünglichen Tabelle und der eingefügten beiden Zeilen sich decken, d. h. ob die richtigen Spalten einander zugeordnet wurden.

(12) Datentabelle nach den Spalten sortieren. Bei Calc geht das wie folgt: Bereich markieren, » *Daten*, » *Sortieren*, dort zum Reiter Einstellungen gehen und unten die Richtung angeben, d. h. die Voreinstellung Von oben nach unten (Zeilen sortieren) umstellen auf Von links nach rechts (Spalten sortieren). Dann zurück zum Reiter Sortierkriterien und dort die richtige Zeile mit den Scores auswählen sowie (wie gewünscht) die Reihenfolge absteigend oder aufsteigend auswählen. Mit OK bestätigen.

(13) Fertig.

(14) Sie können Ihre Tabelle nun noch etwas leserfreundlicher aufbereiten. So ist es z. B. bei großen Tabellen nützlich, wenn man die Spalten- und Zeilennamen an beiden Rändern lesen kann, d. h. die Gräbernamen am rechten und am linken Rand stehen, die Typnamen in der Kopf- und in der Fußzeile. Mit Copy & Paste ist das schnell erledigt. Da die Scores Informationen zu den Abständen der Gräber und Typen enthalten, lösche ich diese Zeile und Spalte nicht, sondern erhalte mir und den Lesern diese nützliche Information. Mit persönlich fällt das Lesen der Zellenhäufigkeiten leichter, wenn die Ziffern in den Zellen zentriert sind (Bereich markieren, Zellen formatieren, Ausrichtung: zentriert). Bei wirklich großen Tabellen kann es helfen, wenn z. B. alle 10 Zeilen eine Zeile eingefügt wird, die später im Druckbild eine waagerechte bzw. senkrechte gestrichelte Linie ergibt.

Für kleine Tabellen mit wenigen Zeilen und Spalten bedeutet dieser semi-automatische Sortierungsprozess einen allzu hohen Aufwand, hier geht das in PAST mögliche händische Sortieren erheblich schneller. Für große Tabellen ist der skizzierte Weg nützlich, weil er nach

kurzer Einübung recht schnell von statten geht und letztlich erheblich weniger fehleranfällig ist als das händische Sortieren.

Entsprechend dem Regelfall wurde hier nach Achse 1 sortiert. Wer im Lauf einer Analyse eine mögliche inhaltliche Dimension in den Achsen 2 oder 3 erkennt, kann auf diesem Weg die Ausgangstabelle selbstverständlich auch nach anderen Achsen als Achse 1 sortieren.

## 19 "Listen" und Warum eine CA mit R?

Wir haben in dieser Einführung bislang mit einer Eingabe der Daten als Tabelle gearbeitet. Sobald das bearbeitete Thema umfangreicher wird und die Tabelle den Umfang von ein bis zwei Bildschirmseiten überschreitet, wird dies mühsam und fehleranfällig. Fehleranfällig, weil man ja stets die eine Zelle treffen muss, d. h. in der richtigen Zeile die richtige Spalte treffen muss. Ja, man kann sich z. B. in LO Calc helfen, indem man die 1. Spalte und die Kopfzeile fixiert… Dennoch: es wird mühsam. Nicht zuletzt: diese Tabellen sind ziemlich leer, d. h. vor allem mit Nullen besetzt. Selbst unser kleines Beispiel mit 10 Gräbern und 10 Typen beinhaltet nur 28 besetzte Zellen, von 100 vorhandenen. Um einmal die Größe echter Forschungsprobleme zu skizzieren: Meine Seriation der merowingerzeitlichen Männergräber vom Niederrhein umfasst 316 Gräber und 158 Typen (d. h. 49.928 Zellen) und hat im Druck als Beilage eine Größe von 80 x 40 cm (Siegmund 1998): Aufgaben dieser Art sind am Bildschirm kaum zu überschauen. Klarstellend: Kollegen haben durchaus größere Tabellen erfolgreich bearbeitet.

Für dieses praktische Problem hatten die Programmautoren von WinBASP (und ihm folgend WinSERION) die Dateneingabe und -verwaltung als Liste eingeführt: Typ xyz kommt vor in Grab abc, def, jkl, usw. Damit wird der Umstand abgebildet, dass die Informationen in der Regel beim Bearbeiter auch so (d. h. als Liste) vorliegen und erst von Listen aus in eine Tabellenform gebracht werden, und dass die Listen uns die Arbeit mit den sehr vielen unbesetzten Zellen ("Nullen") der Tabellen ersparen. Praktiker, die Erfahrung mit großen Datensätzen haben, wissen, dass die Kontrolle der Daten, Korrekturen und Ergänzungen via Listen erheblich einfacherer und vor allem fehlersicher sind, denn bei der Eingabe über Tabellen.

Das hier einführend benutzte PAST enthält alle notwendigen Funktionen zum Durchführen einer CA und verwandter Verfahren. Die Handhabung von PAST ist schnell erlernt, so dass man sich als Anwender bald wieder auf die inhaltliche Arbeit konzentrieren kann. Daher empfehle ich Anfängern, außer man ist bereits R-Nutzer, Seriationsprojekte mit Hilfe von PAST durchzuführen. Außer, man bearbeitet große Datensätze. Unter "groß" verstehe ich Datensätze, die als Tabelle deutlich mehr als eine Bildschirmseite füllen. Weil dann die Dateneingabe und -verwaltung als Liste effizienter und sicherer ist, und dies ist mit **R** möglich, wie im Folgenden dargelegt wird.

**R** ist ein mächtiges, kostenloses und quelloffenes Statistikprogramm, oder besser: Datenverwaltungs- und Statistik-System (R Core Team 2022). Es hat sich seit seinem Start 1992 vor allem im vergangenen Jahrzehnt zum Quasi-Standard unter Statistik-Profis entwickelt (Muenchen 2022). Aber: das Erlernen von **R** ist aufwändig und steht u.a. für

Coden, d. h. ein Arbeiten mit Kommandozeilen statt grafischen Benutzerüberflächen. Gewiss, auch für **R** gibt es (gute!) grafische Benutzeroberflächen ("GUI", *graphical user interface*). Für zwei von ihnen, den R-Commander und Jamovi, stehen sogar Plugins bereit, mit denen eine CA gerechnet werden kann (R-Commander: Plugin FactoMineR; Jamovi: Plugin Multivariate Exploratory Data Analysis); beide Plugins nutzen das R-Paket FactoMineR. Aber: diese GUIs erwarten als Eingabe eine Datentabelle. Die listenweise Eingabe der Daten, die hier der Grund für den Einsatz von **R** ist, muss anderweitig erfolgen oder vorgelagert erfolgen, d. h. bevor man mit einer GUI weiterarbeiten kann. Übersetzt: die Funktionalität, für die m. E. die Abwendung von PAST sinnvoll ist, erfordert in **R** sogleich den größeren Lernschritt, nämlich das Coden statt dem Arbeiten mit einer R-GUI.

Das Folgende ist kein R-Kurs! Es geht vielmehr zielgerichtet um das Durchführen einer CA mit **R**, d. h. alles Darüberhinausgehende wird nicht vermittelt. Wer im Feld Archäologie & Statistik weiterkommen und Profi-Niveau erreichen will, sollte **R** erlernen – aber das ist nicht Thema dieses Praxisleitfadens. Die vorliegende Einführung enthält nun auch kommentierten und erklärten R-Code, so dass Anwender mit Copy & Paste erbeiten können und in diesem Code nur das Nötige auf Ihre Arbeit anpassen müssen. Wer sich gründlicher mit R beschäftigen möchte, konsultiere z. B. Siegmund (2020) oder Kabacoff (2022).

## 20 R: Einrichten des Arbeitsplatzes

Für ein effizientes Arbeit mit **R** benötigen wir drei Komponenten: R, RStudio und RTools, sowie R-Pakete. Wir beginnen mit den drei erstgenannten Komponenten.

### R

Das Programm **R** steht kostenlos auf CRAN "The Comprehensive R Archive Network" zur Verfügung, dort: https://cran.r-project.org/ Die aktuellen Versionen werden oben auf der Startseite angeboten, in meinem Fall ist das "Download R for Windows". Die Folgeauswahl ist unter "*base*" die Option "*install R for the first time*", dann "*Download R-4.2.2 for Windows*" (Stand Jan. 2023). Es werden ca. 76 MB heruntergeladen für ein 64-bit-Windows Betriebssystem. Für ein 32-bit-Windows oder andere Betriebssysteme findet man die entsprechenden Optionen auf der gen. Website leicht. Nun im eigenen Download-Ordner auf die Installationsdatei gehen, klicken, usw. Die von **R** gesetzten Voreinstellungen bei den Abfragen während der Installation sind in den allermeisten Fällen passend, es besteht kein Anlass, etwas zu ändern. Nach der Installation umfasst der Programmordner etwa 164 MB, womit wir abschätzen können, wie viel Platz auf Ihrer Festplatte mindestens frei sein sollte.

Sofern Sie die Voreinstellungen belassen haben, finden Sie nun auf Ihren Desktop das Symbol für **R**. Anklicken zum Ausführen, um zu prüfen, ob alles korrekt verlaufen ist. Der Bildschirm sollte nun ein Fenster zeigen, die "R Console" (**Abb. 20**) In diesem Fenster sehen Sie blaue Schrift und zuunterst einen liegenden, nach rechts weisenden roten Pfeil: hinter diesem Pfeil

erwartet **R** Ihre Eingaben. Vorschlag: Sie tippen hinter dem Pfeil nun *citation()* ein (dann ENTER), was abruft, wie man diese R-Version zitieren sollte. Kopieren Sie sich dieses Zitat und speichern es in Ihrer Literaturliste für das anstehende Projekt ab - denn man sollte **R** regulär zitieren, so wie man auch Aufsätze, Bücher etc. zitiert, die man genutzt hat. Nun tippen Sie hinter dem roten Pfeil: *q()* - womit man die R Console regulär beendet. Die Abfrage "Workspace sichern" beantworten Sie mit "Nein". **R** ist damit installiert und arbeitsfähig.

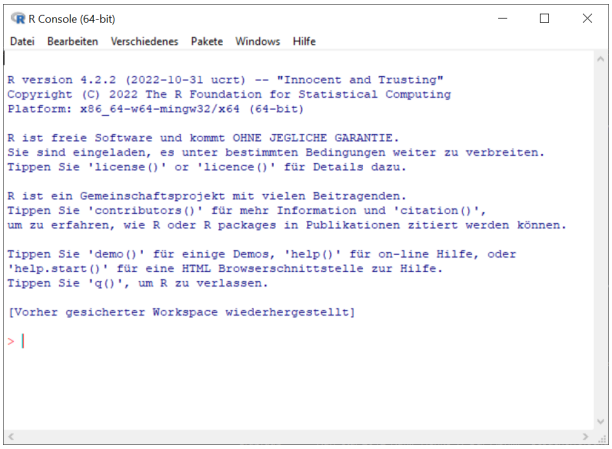



**Abb. 20** Die R-Console unmittelbar nach dem Start.

### RStudio

Die meisten Anwender nutzen nicht das rohe **R** (das wir gerade gesehen haben), sondern arbeiten mit RStudio, was nicht als graphische Benutzeroberfläche gilt, sondern als Entwicklungsumgebung. Zwar muss man **R** auch unter RStudio coden, aber RStudio bietet sehr viele Unterstützungen dafür. RStudio ist ein kostenloses, offenes Produkt der Firma "posit" und findet sich dort: https://posit.co/ Hinter dem Schalter "Download" wartet eine neue Seite mit Auswahloptionen. Unsere Auwahl ist "RStudio Desktop Free". Die Bezahl-Alternative "Pro" erbringt das gleiche RStudio wie die Free-Version, aber zusätzlich professionelle Beratung, Cloud-Space usw. - für unsere Zwecke ist "Free" vollends hinreichend. Da wir soeben ein frisches **R** installiert haben, setzen wir beim Schritt 2 ein: "Install RStudio Desktop". Alles Weitere wie gewohnt: im Download-Ordner die Installationsdatei (ca. 200 kb) klicken, … Auch hier sollte man die Abfragen gemäß der Voreinstellungen annehmen. Das installierte RStudio braucht ca. 790 MB Platz.

Nach dem Starten des Programms RStudio öffnet sich ein drei- oder vierteiliges Fenster (**Abb. 21**), in dem man links bereits die aktive R Console sieht. Kurz zu diesen vier Fenstern: RStudio zeigt "eigentlich" vier verbundene Teilfenster. Oben hat RStudio zwei allgemeine Bedienleisten, darunter vier Teilfenster. Jedes dieser vier Teilfenster hat oben einen grauen Balken mit ein paar Reitern, und ganz rechts in grau ein oder zwei Symbole, welche eben diese Fenster darstellen. Sieht man links nur ein (großes) Teilfenster, klicke man auf dieses graue

Symbol. Es faltet sich das zweite linke Teilfenster auf. Sie sollten jetzt (oder bereits zuvor) sehen:

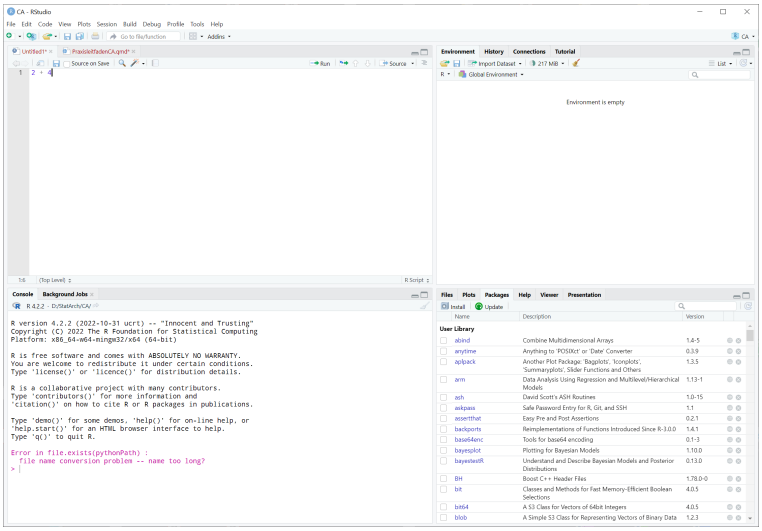


**Abb. 21** Die Standardansicht von RStudio mit vier Fenstern.

Die Teilfenster kann man, wie gerade vorgenommen, auf- oder zuklappen. Man kann sie relativ zueinander verkleinern oder vergrößern, indem man mit der Maus auf den grauen Mittelbalken geht und dort "zieht".

- Das Teilfenster links oben ist das RStudio-Fenster, in dem man R-Code schreibt, kommentiert, ausführt (rechts oben, "Run" mit grünem Pfeil), speichert etc.
- Das Teilfenster links unten ist die R-Console, d. h. das aktuell laufende **R**, wie wir es schon unmittelbar nach der Installation von **R** testweise aufgerufen hatten. RStudio erlaubt es, auch dort an dem liegenden Pfeil direkt Befehle einzugeben.
- Das Fenster rechts oben mit "Environment", "History", "Connections" und "Tutorial" enthält das Besagte und hilft Ihnen beim Arbeiten, insbes. Daten zu importieren und den Überblick über Ihre Daten zu bewahren.
- Das Fenster unten rechts mit "Files", "Plots", "Packages", "Help"… zeigt u.a. hinter "Plot" die grafischen Ausgaben, die Sie mit R angelegt haben , vor allem aber auch unter "Packages" eine Liste der installierten R-Pakete (dazu gleich mehr). "Help" versteht sich von selbst und liefert über das Suche-Fenster viele Informationen insbes. zum Code.

Weiteres zu RStudio folgt sogleich, doch zunächst installiere man noch eine dritte Komponente:

### RTools

Die Grundausstattung von **R** kann durch zahlreiche "Pakete" (Unterprogramme) erweitert werden. Was wir auch brauchen, denn eine Korrespondenzanalyse gehört nicht zur Grundausstattung von **R**. In der Regel erhält man diese Pakete als "binaries", als fertige, lauffähige Programme. Bisweilen aber sind diese Pakete noch nicht kompiliert / "übersetzt", sondern werden als Programm heruntergeladen und dann erst auf Ihrem Rechner automatisch kompiliert. Auf Windows-PCs braucht man für dieses automatische Kompilieren das Programm RTools - sonst gibt es überraschende Fehlermeldungen. Um diese Irritationen gleich von Anbeginn an zu vermeiden, installieren wir also als dritte Komponente RTools. Das kostenlose RTools gibt es dort: https://cran.r-project.org/bin/windows/Rtools/ Da wir soeben **R** und RStudio frisch installiert haben, ist die dort zuoberst angeführte, jüngste Ausgabe von RTools das Richtige für uns.

Auf der (leicht unübersichtlichen) Folgeseite wählen wir den gut sichtbaren Link zu "Rtools43_installer" (ca. 475 MB) und installieren diesen dann wie üblich aus unserem Download-Ordner. Auch hier gilt: Voreinstellungen beibehalten. Fertig, denn mit RTools arbeiten wir nicht aktiv selbst, sondern **R** und RStudio benötigen RTools gelegentlich und sprechen es von sich aus automatisch an. "rtools43" nimmt nach der Installation ca. 3 GB Platz ein, also eher viel. Wem das Probleme bereitet, verzichtet auf RTools, kann dann allerdings nur Pakete laden, die als Binaries zur Verfügung stehen.

Übrigens: es ist ein guter Zeitpunkt, jetzt die nicht mehr benötigten Installer-Dateien in Ihrem Download-Ordner zu löschen.

### R-Pakete

Mit der Grundausstattung von **R** kann man schon sehr viel Datenverwaltung, Statistik und Grafik machen. Doch auf CRAN stehen zahlreiche "Pakete" bereit, mit denen man die Grundfunktionen von **R** erweitern kann. Anfang 2023 waren es über 18.500 Pakete, die Liste findet sich dort: https://cran.r-project.org/ und zwar unter (linke Spalte) "Packages" und "sorted by name". Ein weiteres wichtiges Pakete-Lager ist "Bioconductor" mit Anfang Januar 2023 über 2.100 weiteren R-Paketen; näheres dort: https://www.bioconductor.org/

Kein Stress, RStudio hilft beim Installieren und macht es leicht. Klug durchdacht, aber anfangs verwirrend: Diese Unterprogramme gibt es für **R** in zwei Zuständen: installiert und aktiviert. Zunächst wird ein R-Paket installiert, d. h. auf Ihrem Rechner eingerichtet und bereitgestellt. Das kann man in der R-Console mit "install(*paketname*)" machen, bequemer ist es, die Funktionen von R-Studio dafür zu nutzen. Um Rechenspeicher zu sparen, werden jedoch beim Starten von **R** oder RStudio keinesfalls alle installierten Pakete auch aktiviert. Daher versetzt man bei Bedarf mit "library(*paketname*)" eines der installierten Zusatzpakete in einen aktiven Zustand - auch das geht per RStudio sehr bequem. Wir kommen in Kap. 23

zur Praxis dieses Vorgehens. Eine nicht seltene Fehlermeldung bei R beruht darauf, dass man ein Paket zwar installiert, aber nicht aktiviert hat …

Wir wollen zu Übungszwecken, aber auch, weil es inhaltlich sinnvoll ist, sogleich ein R-Paket installieren: readxl - ein Paket, das R und RStudio darin unterstützt, Excel-Dateien zu importieren. Wiewohl es andere, automatisiertere Wege gibt, wollen wir diese Installation "händisch" vornehmen, einfach, um's einmal gemacht zu haben. Unter RStudio gibt es zwei Ansatzpunkte: (a) In der Bedienleiste ganz oben: » *Tools*, dann *Install packages, …* oder (b) im RStudio-Teilfenster rechts unten, Reiter *Packages*, dann *Install*, … Es öffnet sich in beiden Fällen ein Popup-Fenster, in dem Sie in der Mitte einfach "readxl" eingeben, wobei via Autovervollständigung der Name das Pakets schon früh auftaucht. Namen vollends und korrekt eingeben (übernehmen), unten den "Install"-Button drücken, fertig. In der R-Console (links unten) gibt es dann verschiedene Meldungen, die den Installationsprozess dokumentieren.

Alle auf Ihrem PC bereits installierten (sic) R-Pakete sind in alphabethischer Reihenfolge aufgelistet, wenn man im RStudio-Teilfenster rechts unten auf den Reiter "Packages" klickt. Wenn man in das leere quadratische Kästchen vor dem Paketnamen klickt und ein Häkchen setzt, wird das jeweilige Paket auch aktiviert, d. h. der Befehl "library(*paketname*)" ausgeführt. Doch das wollen wir jetzt nicht tun, weil dieses Aktivieren meist anderweitig erfolgt.

Hinweis: Sie arbeiten vermutlich jetzt mit einer ganz frischen Version von **R**. Aber **R** ist sehr lebendig, viele Pakete werden von Zeit zu Zeit erneuert. Unter dem Reiter "Packages" finden Sie auch den Knopf "Update". Damit werden alle (!) installierten Pakete auf Aktualität überprüft. Ein Popup-Fenster listet alle Pakete, für die ein Update bereitsteht. Wahlweise kann man dann die zu erneuernden Pakete gezielt anwählen, oder einfach alle Pakete erneuern. Wiederum werden in der R-Console die Protokolle dieses Prozesses ausgegeben.

### RStudio: ein Projekt einrichten

Abschliessend noch: ein Projekt einrichten. Worum geht es? Es geht um das Ordnung-Halten und um Übersicht. Ich empfehle Ihnen: für jedes Ihrer Vorhaben einen eigenen Ordner (ggf. samt Unterordnern) auf Ihrem PC. Sofern Sie nicht regelmäßig und mind. wöchentlich in die Cloud sichern, erlaubt Ihnen dies auch eine gute (Teil-) Sicherung Ihrer Daten: das aktive Projekt (d. h. diesen 1 Ordner) täglich auf ein externes Medium.

RStudio unterstützt Sie dabei. Zunächst der konventionelle Weg:

```
getwd()     # das aktuelle Arbeitsverzeichnis abfragen
```

```
[1] "D:/StatArch/CA"
```

```
#
# setwd("D:/StatArch/CA") # für R & RStudio ein Arbeitsverzeichnis absolut setzen, d.h. mi
#
# oder ein Arbeitsverzeichnis relativ vom aktuellen Verzeichnis aus setzen:
# setwd("Unterverzeichnis24/undnochtiefer")
```

Beachte: die entsprechenden Befehle "setwd…" sind hier mit einem anführenden "#" = Kommentarzeichen absichtlich de-aktiviert worden, damit sie nicht ausgeführt werden, Sie den Code per Copy&Paste aber verwenden können.

Der bessere Weg ist es, via RStudio ein "Projekt" anzulegen. Der entsprechende Schalter sitzt in RStudio ganz oben außen rechts. Nach Draufklicken bietet das Popup-Fenster die Optionen, ein neues Projekt anzulegen oder ein bestehendes Projekt zu öffnen, zudem zeigt es alle ihm aktuell bekannten Projekte an, die Sie bereits vereinbart haben. Sie können also an dieser Stelle auch schnell zwischen verschiedenen Projekten hin- und herschalten. Wenn Sie ein neues Projekt anlegen, können sie es in einen bestehenden Ordner legen oder von RStudio einen neuen Ordner anlegen lassen. Ab dann arbieten Sie in diesem Projekt, d. h. **R** legt per Voreinstellung alle Dateien dorthinein ab resp. liest sie von hier. Im RStudio-Teilfenster rechts unten hinter dem Reiter "Files" sehen Sie dann das Dateiverzeichnis Ihres aktuellen Projekts / Projektordners. [ Das alltägliche Sichern dieses Verzeichnisses müssen Sie allerdings selbst durchführen. ]

Packen Sie die Übungsdaten, die zu diesem Leitfaden dazugehören, in Ihren Projektorder. Dann wird RStudio sie ohne Pfad-Angaben finden, und sie im Fenster unten rechts hinter dem Reiter "files" auch auflisten.

Mit diesen Schritten ist das Einrichten von R, RStudio und eines Arbeitsplatzes abgeschlossen und wir können an die inhaltliche Arbeit gehen.

### Hinweis zur Nutzung dieser Einführung

Die nachfolgend in den Text eingebetteten Code-Blöcke sind dazu gedacht, von Ihnen - ggf. mit Anpassungen - per Copy & Paste genutzt werden zu können. Aus der gedruckten Ausgabe heraus ist das natürlich nicht möglich. Zusätzlich stelle ich daher den Text als *.html-Ausgabe zur Verfügung. Geht man dort mit der Maus in die rechte obere Ecke eines grau unterlegten Codeblockes, wird ein Zeichen (ca. Notizblock) sichtbar. Mit einem Mausklick auf dieses Zeichen wird der gesamte Codeblock in den Zwischenspeicher kopiert. Von dort in das linke obere Fenster von RStudio per Klick einfügen: der Code kann nun auf Ihrem PC ausgeführt werden. Empfehlung: (a) sich diesen Code selbst auskommentieren mit anführendem "#", (b) später ggf. Überflüssiges und fehlerhafte Versuche streichen und dann (c) diesen Ihren Code unter RStudio speichern. So können Sie ihre erfolgreiche CA später wieder laden und erneut ausführen, ev. mit verbesserter Datentabelle.

Die Code-Blöcke sind hier so eingestellt, dass die von **R** ausgegebenen Ergebnisse dem jeweiligen Code-Block nachgestellt werden. Das bläht den Text etwas auf und ist für Diejenigen überflüssig, die selbst am PC sitzen und den Text mit **R** nacharbeiten. Es ermöglicht aber blätternden Lesern, die erst einmal Eindrücke und einen Überblick gewinnen wollen, eine Anschauung dessen, was auf sie zukommt.

Dieser Text wurde mit "Quarto" geschrieben, dem aktuellen RMarkdown-Editor von RStudio. Um Sie zu ermuntern, sich zusätzlich auch mit Quarto vertraut zu machen, stelle ich den Text dieser Einführung auch als Quarto-Dokument zur Verfügung (*.qmd). Diese Art des Schreibens in RMarkdown oder Quarto mit enger Verknüpfung von Text und lauffähigem Code ist gerade auch während des Forschungsprozesses nützlich, weil das Ergebnis übersichtlicher ist als komplexer und langer in **R** kommentiert Code und man so seinen eigenen Code und eigenen Text schrittweise entwickeln kann, bis ein sauberes Produkt fertig ist. Quarto erlaubt u. a. auch die Ausgabe als *.docxs, man kann das Schreiben in Quarto also später problemlos verlassen und in gewohnten Systemen weiterarbeiten.

## 21 Tabellen in R: data.frame und data.matrix

**Zu den Kap. 21, 22 und 23:**

- Sofern es Ihnen "nur" darum geht, mit einer vorhandenen Tabelle (z. B. den Übungsdateien) in **R** eine CA zu rechnen und Ihnen die Themen "Data Frame" / "Matrix" sowie "Tabellen in **R**" und "Listeneingabe von Daten" unwichtig sind, können Sie das unmittelbar Folgende überspringen, gleich Ihre Daten als Matrix auf den Namen "dm" einlesen und mit Kap. 24 fortfahren.
- In Kap. 21 und 22 lernen wir das Arbeiten mit Tabellen und Listen etwas ausführlicher kennen, damit das Nachfolgende nicht nur "nachgekocht", sondern auch verstanden werden kann. Wer indes sogleich die Dateneingabe als Liste vornehmen möchte, sich für die vorgelagerten Details nicht interessiert, vielmehr möglichst bald auf Basis einer Eingabe-Liste eine CA durchführen möchte, überspringe Kap. 21 und 22 und starte mit Kap. 23, das zeigt, wie man schnell von Listen zur nötigen Eingabematrix kommt.

Ein fertige Datentabelle in **R** einlesen ist sehr einfach. Man kann z. B. in RStudio im Teilfenster rechts oben hinter dem Reiter "Environment" in der Leiste darunter auf "Import Dataset" klicken. Das Popup-Fenster listet Möglichkeiten, und z. B. mit "From Excel" wird das Paket readxl aktiviert und vollzieht über ein eigenes Popup-Fenster auf selbsterklärendem Weg zum Datenimport. Weil wir aber für die CA und die listenweise Eingabe von Daten den Umgang von **R** und seine Tabellen besser verstehen müssen, erlernen wir im Folgenden den Aufbau von Daten in **R** von Grund auf. Braucht etwas mehr Zeit als der zuvor gezeigte Weg, ist aber notwendig und nützlich.

**R** kennt für die Daten (Variablen) die üblichen Datenformate, insbes. Zahl (*numeric*), Zeichen(-kette) (*character*) und logisch (*TRUE, FALSE*). Vor allem aber "denkt" R in

Vektoren, das sind Ketten von gleichartigen Daten. Die Vektoren entsprechen dem, was in den von LO-Calc oder MS-Excel gewohnten Tabellen die Spalten sind. In **R** kann man die Datenketten aus einzelnen Werten eingeben und zu Vektoren machen, man kann Tabellen mehreren Vektoren zusammengesetzt werden.

**R** kennt zwei unterschiedliche Tabellen: Dataframe und Matrix. Ein Dataframe ist eine Tabelle, deren Spalten unterschiedliche Arten von Variablen enthalten, also z. B. eine Spalte mit Zahlen, eine Spalte mit Buchstaben usw. Eine Matrix ist eine Tabelle, deren Spalten aus gleichartigen Variablen bestehen, z. B. durchweg aus Zahlen oder durchweg aus Buchstaben. In den folgenden Code-Blöcken setzen wir dies in Beispiele um, wobei wir zunächst einen Dataframe bauen. Im ersten Abschnitt erzeugen wir drei Spalten mit den Namen "Zahlen", "Buchstaben" und "Booleans" mit je vier Werten und schauen uns die resultierende Tabelle an. Bitte nehmen Sie auch wahr, dass im RStudio Teilfenster oben rechts, hinter dem Reiter "Environment" die "Data" angezeigt werden; nach Ausklappen von "df" erscheinen weitere Informationen.

## 21.1 "Dataframe"

Mit der folgenden Anweisung bauen wir einen Dataframe, bestehend aus drei Spalten mit je vier Fällen = Zeilen. Dabei ist "df" der Name, den wir unserem Dataframe selbst geben.

```
# Dataframe df bauen:
df <- data.frame(Zahlen=11:14, Buchstaben=letters[1:4],
                 Booleans=(1:4)>2)
# data.frame ansehen:
df
```

```
  Zahlen Buchstaben Booleans
1     11          a    FALSE
2     12          b    FALSE
3     13          c     TRUE
4     14          d     TRUE
```

Sie können den Code-Block (*chunk*) jeweils copy-pasten in das linke obere Fenster von RStudio und dort ausführen. Zum ersten Lernen besser jeweils 1 Befehl und dann nacheinander ausführen. Wie? Mit dem Cursor in die jeweilige Zeile gehen und dann in der Leiste unmittelbar über dem Editierfenster rechts auf "Run" klicken. Nach dem "Run" sehen Sie im RStudio Teilfenster unten links, d. h. in der R-Console, die Ausgabe.

Nun lernen wir noch zwei weitere Optionen kennen, sich einen Dataframe im Überblick anzuschauen. Die folgende Art entspricht dem, was RStudio im Teilfenster rechts oben auch unter "Envronment" als "Data" anzeigt:

```
  str(df)
```

```
'data.frame':   4 obs. of  3 variables:
 $ Zahlen    : int  11 12 13 14
 $ Buchstaben: chr  "a" "b" "c" "d"
 $ Booleans  : logi  FALSE FALSE TRUE TRUE
```

… oder als zusammenfassende Statistik:

```
  summary(df)
```

```
     Zahlen       Buchstaben         Booleans
 Min.   :11.00   Length:4           Mode :logical
 1st Qu.:11.75   Class :character   FALSE:2
 Median :12.50   Mode  :character   TRUE :2
 Mean   :12.50
 3rd Qu.:13.25
 Max.   :14.00
```

Sodann kan man aus einer Tabelle auch einzelne Werte oder Werteketten auslesen, indem man sie “indiziert”. Wie, zeigt der folgende Codeblock:

```
  df[4,1]  # liest der Wert in der 4. Zeile und 1. Spalte aus
```

```
[1] 14
```

```
  #
  df[ ,2]   # liest alle Werte in der 2. Spalte aus
```

```
[1] "a" "b" "c" "d"
```

```
  #
  df[2, ]  # liest alle Werte in der 2. Zeile aus
```

```
  Zahlen Buchstaben Booleans
2     12          b    FALSE
```

Wichtig sind die eckigen Klammern, die nach dem Schema [*Zeile* Komma *Spalte*] aufgebaut sind. Setzt man sowohl für *Zeile* als auch *Spalte* einen Wert ein, wird der Inhalt einer Zelle ausgegeben. Setzt man keinen Wert ein bei *Zeile*, wird die ganze Datenzeile ausgegeben, usf.

In den folgenden Code-Blöcken lernen wir exemplarisch weitere Möglichkeiten kennen. Während des Übens werden Sie denken "warum so kompliziert: in Excel oder PAST kann ich das doch einfach per Mausklick?" Ja. Aber für das Kommende ist es wichtig, dass es auch ohne Mausklick in eine Tabelle geht und wir hier exemplarisch die Möglichkeiten kennen lernen. Zunächst informieren wir uns wieder über den aktiven Dataframe.

```
nrow(df)      # Zeilen eines data.frames zählen
```

```
[1] 4
```

```
#
ncol(df)      # Spalten eines data.frames zählen
```

```
[1] 3
```

```
#
colnames(df)  # Alle Spaltennamen des data.frames ausgeben lassen
```

```
[1] "Zahlen"     "Buchstaben" "Booleans"
```

Nun greifen wir verändernd in die Datentabelle ein, zunächst auf die Spalten- und Zeilennamen::

```
colnames(df)[2] <- "Zeichen"     # den Spaltenkopf der 2. Spalte ändern
colnames(df)
```

```
[1] "Zahlen"   "Zeichen"  "Booleans"
```

```
# Beachte: die 2. Spalte, bisher "Buchstaben", heisst nun "Zeichen".
#
# Wir geben den Zeilen des data.frames einen Namen, i. e. die Zeilen umbenennen:
rownames(df)   # vorher
```

```
[1] "1" "2" "3" "4"
```

```
rownames(df)[1:4] <- c("Z1","Z2","Z3","Z4")
rownames(df)   # nachher
```

```
[1] "Z1" "Z2" "Z3" "Z4"
```

Nun ändern wir Zellen, d. h. Inhalte:

```
# Zuerst: einen einzelnen Wert ausgeben lassen, indem man ihn mit Zeilen- und Spaltenname
df["Z2","Zeichen"]
```

```
[1] "b"
```

```
#
# Einen Wert ändern, z.B. zur Fehlerkorrektur:
df["Z2","Zeichen"] <-  "xx"
df
```

```
   Zahlen Zeichen Booleans
Z1     11       a    FALSE
Z2     12      xx    FALSE
Z3     13       c     TRUE
Z4     14       d     TRUE
```

```
#
# Der Tabelle eine neue Spalte mit Zahlen hinzufügen:
df$NEU <- c(99,98,97,96)
df
```

```
   Zahlen Zeichen Booleans NEU
Z1     11       a    FALSE  99
Z2     12      xx    FALSE  98
Z3     13       c     TRUE  97
Z4     14       d     TRUE  96
```

```
# Aus der Tabelle eine Spalte entfernen (löschen):
df$Booleans <- NULL
df
```

```
   Zahlen Zeichen NEU
Z1     11       a  99
Z2     12      xx  98
Z3     13       c  97
Z4     14       d  96
```

BAUSTELLE: Dataframe intern sichern (und wieder einlesen)

```
save(df, file="dateiname.RData")
```

### 21.2 "Matrix"

Die Datentabelle, die wir im ersten Teil unter PAST bei der Dateneingabe benutzt hatten, bestand jedoch nicht aus Spalten mit unterschiedlichen Arten von Daten, sondern ausschließlich aus Zahlen. Nur die Zeilennamen und die Spaltenköpfe waren Zeichenketten. Nach **R** übersetzt: es handelte sich um eine data.matrix. Daher bauen wir im nächsten Übungsschritt eine Daten-Matrix. Unsere Übungsmatrix, die wir kurz mit "dm" benennen, soll 4 Zeilen und 6 Spalten haben, die alle den Wert Null haben:

```
dm <- matrix(data=0, nrow=4, ncol=6)
dm
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    0
[2,]    0    0    0    0    0    0
[3,]    0    0    0    0    0    0
[4,]    0    0    0    0    0    0
```

Wie zuvor beim Dataframe informieren wir uns über den Inhalt dieser Matrix:

```
nrow(dm)    # Anzahl Zeilen?
```

```
[1] 4
```

```
ncol(dm)    # Anzahl Spalten?
```

```
[1] 6
```

```r
dim(dm)     # Dimension, d.h. Anzahl Zeilen und Spalten
```

```
[1] 4 6
```

```r
length(dm)  # Anzahl Felder/Zellen
```

```
[1] 24
```

Ausgabe der Zeilen- und Spaltensumme:

```r
rowSums(dm)   # Summe über alle Zeilen (großes "S" beachten):
```

```
[1] 0 0 0 0
```

```r
#
colSums(dm)   # Summe über alle Spalten:
```

```
[1] 0 0 0 0 0 0
```

Nun geben wir den Zeilen und Spalten Namen:

```r
colnames(dm) <- c("Typ1", "Typ2", "Typ3", "Typ4", "Typ5", "Typ6")
rownames(dm) <- c("Grab1", "Grab2", "Grab3", "Grab4")
dm
```

```
      Typ1 Typ2 Typ3 Typ4 Typ5 Typ6
Grab1    0    0    0    0    0    0
Grab2    0    0    0    0    0    0
Grab3    0    0    0    0    0    0
Grab4    0    0    0    0    0    0
```

Nun geben wir anhand der Zeilen- und Spaltennamen in drei der Zellen Werte ein:

```r
dm["Grab1","Typ1"] <- 11
dm["Grab2","Typ2"] <- 22
dm["Grab3","Typ3"] <- 33
```

```
dm
```

```
      Typ1 Typ2 Typ3 Typ4 Typ5 Typ6
Grab1   11    0    0    0    0    0
Grab2    0   22    0    0    0    0
Grab3    0    0   33    0    0    0
Grab4    0    0    0    0    0    0
```

Da bei einer Matrix alle Spalten von der gleichen Art sind, also alle entweder *numeric* oder *character* oder *Boolean*, kann eine Matrix auch ans Ganze modifiziert werden. Im folgenden Beispiel multiplizieren wir alle Fälle mit 3:

```
dm <- dm * 3
dm
```

```
      Typ1 Typ2 Typ3 Typ4 Typ5 Typ6
Grab1   33    0    0    0    0    0
Grab2    0   66    0    0    0    0
Grab3    0    0   99    0    0    0
Grab4    0    0    0    0    0    0
```

Wie beim Dataframe können Werte gezielt aus der Matrix ausgelesen werden. Da wir die Spalten und Zeilen benannt haben, kann die nötige Indizierung über die Spalten- und Zeilennamen geschehen:

```
dm[,"Typ3"]   # alle Werte einer Spalte ausgebe
```

```
Grab1 Grab2 Grab3 Grab4
    0     0    99     0
```

```
#
dm["Grab1", ] # alle Werte einer Zeile ausgeben
```

```
Typ1 Typ2 Typ3 Typ4 Typ5 Typ6
  33    0    0    0    0    0
```

```
#
```

Ausgabe einer Teil-Tabelle:

```
dm[ ,1:3]
```

```
      Typ1 Typ2 Typ3
Grab1   33    0    0
Grab2    0   66    0
Grab3    0    0   99
Grab4    0    0    0
```

```
#
dm[1:3,1:4]
```

```
      Typ1 Typ2 Typ3 Typ4
Grab1   33    0    0    0
Grab2    0   66    0    0
Grab3    0    0   99    0
```

"Gewichten", d. h. einzelne Typen weniger wichtig / häufig machen:

```
dm   # vorher
```

```
      Typ1 Typ2 Typ3 Typ4 Typ5 Typ6
Grab1   33    0    0    0    0    0
Grab2    0   66    0    0    0    0
Grab3    0    0   99    0    0    0
Grab4    0    0    0    0    0    0
```

```
dm[,"Typ3"] <- dm[,"Typ3"]/4
dm   # nachher
```

```
      Typ1 Typ2  Typ3 Typ4 Typ5 Typ6
Grab1   33    0  0.00    0    0    0
Grab2    0   66  0.00    0    0    0
Grab3    0    0 24.75    0    0    0
Grab4    0    0  0.00    0    0    0
```

Letzter Teil dieser Sequenz: eine Matrix "transponieren", d. h. Zeilen und Spalten vertauschen:

```
t(dm)
```

```
     Grab1 Grab2 Grab3 Grab4
Typ1    33     0  0.00     0
Typ2     0    66  0.00     0
Typ3     0     0 24.75     0
Typ4     0     0  0.00     0
Typ5     0     0  0.00     0
Typ6     0     0  0.00     0
```

Man beachte: mit "t(dm)" ist die Matrix nur temporär transponiert worden. Wenn wir mit "dm" das Bild der aktuellen Matrix aufrufen, ist diese unverändert. Soll die Transponierung dauerhaft sein, muss die Matrix entsprechend "zugewiesen" werden:

```
dm      # Ist-Zustand
```

```
      Typ1 Typ2  Typ3 Typ4 Typ5 Typ6
Grab1   33    0  0.00    0    0    0
Grab2    0   66  0.00    0    0    0
Grab3    0    0 24.75    0    0    0
Grab4    0    0  0.00    0    0    0
```

```
t(dm)   # Transponierung
```

```
     Grab1 Grab2 Grab3 Grab4
Typ1    33     0  0.00     0
Typ2     0    66  0.00     0
Typ3     0     0 24.75     0
Typ4     0     0  0.00     0
Typ5     0     0  0.00     0
Typ6     0     0  0.00     0
```

```
dm      # Ist-Zustand (unverändert)
```

```
      Typ1 Typ2  Typ3 Typ4 Typ5 Typ6
Grab1   33    0  0.00    0    0    0
Grab2    0   66  0.00    0    0    0
Grab3    0    0 24.75    0    0    0
Grab4    0    0  0.00    0    0    0
```

```
dm <-t(dm)  # Transponieren mit Umspeichern
dm      # Ist-Zustand
```

```
     Grab1 Grab2 Grab3 Grab4
Typ1    33     0  0.00     0
Typ2     0    66  0.00     0
Typ3     0     0 24.75     0
Typ4     0     0  0.00     0
Typ5     0     0  0.00     0
Typ6     0     0  0.00     0
```

## 22 Eingabe einer Tabelle als Liste

Mit dem bislang Erlernten bauen wir eine Matrix auf als Eingabe in die CA. Inhaltlich soll es die Eingabematrix **Abb. 3** sein, die wir im ersten Teil für das Üben mit PAST verwendet haben. Das Ganze erfolgt in zwei Schritten: Zuerst bauen wir eine leere Matrix in der gewünschten Größe auf (hier: 10 x 10 Felder) und füllen alle Zellen mit Null. Im zweiten Schritt werden alle Zellen, die nicht "Null" sein sollen, via Listeneingabe mit der gewünschten Zahl / Häufigkeit gefüllt.

```
dm <- matrix(data=0, nrow=10, ncol=10)  # leere Matrix mit 10 x 10 Feldern
dm
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    0    0    0    0    0    0    0    0     0
 [3,]    0    0    0    0    0    0    0    0    0     0
 [4,]    0    0    0    0    0    0    0    0    0     0
 [5,]    0    0    0    0    0    0    0    0    0     0
 [6,]    0    0    0    0    0    0    0    0    0     0
 [7,]    0    0    0    0    0    0    0    0    0     0
 [8,]    0    0    0    0    0    0    0    0    0     0
 [9,]    0    0    0    0    0    0    0    0    0     0
[10,]    0    0    0    0    0    0    0    0    0     0
```

```
# Denen wir nun Zeilen- und Spaltennamen geben:
# Für die Namen verwenden wir "t..." für Typ / type
# und "f..." für feature / Befund / Grab
#
colnames(dm) <- c("tA", "tB", "tC", "tD", "tE", "tF", "tG", "tH", "tI", "tK")
```

```
rownames(dm) <- c("f1", "f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10")
dm
```

```
    tA tB tC tD tE tF tG tH tI tK
f1   0  0  0  0  0  0  0  0  0  0
f2   0  0  0  0  0  0  0  0  0  0
f3   0  0  0  0  0  0  0  0  0  0
f4   0  0  0  0  0  0  0  0  0  0
f5   0  0  0  0  0  0  0  0  0  0
f6   0  0  0  0  0  0  0  0  0  0
f7   0  0  0  0  0  0  0  0  0  0
f8   0  0  0  0  0  0  0  0  0  0
f9   0  0  0  0  0  0  0  0  0  0
f10  0  0  0  0  0  0  0  0  0  0
```

Nun füllen wir die Daten in die leere Matrix. Das Schema ist das immergleiche: über die Indizierung als [Zeile , Spalte] sprechen wir die einzelne Zelle an, in die ein Wert eingefüllt werden soll, und weisen den entsprechenden Wert zu - in unserem Fall eine “1” oder eine “2”.

```
dm["f1","tA"] <- 2
dm["f1","tB"] <- 1
#
dm["f2","tA"] <- 1
dm["f2","tB"] <- 2
dm["f2","tC"] <- 1
#
dm["f3","tB"] <- 1
dm["f3","tC"] <- 2
dm["f3","tD"] <- 1
#
dm["f4","tC"] <- 1
dm["f4","tD"] <- 2
dm["f4","tE"] <- 1
#
dm["f5","tD"] <- 1
dm["f5","tE"] <- 2
dm["f5","tF"] <- 1
#
dm["f6","tE"] <- 1
dm["f6","tF"] <- 2
dm["f6","tG"] <- 1
```

```
#
dm["f7","tF"] <- 1
dm["f7","tG"] <- 2
dm["f7","tH"] <- 1
#
dm["f8","tG"] <- 1
dm["f8","tH"] <- 2
dm["f8","tI"] <- 1
#
dm["f9","tH"] <- 1
dm["f9","tI"] <- 2
dm["f9","tK"] <- 1
#
dm["f10","tI"] <- 1
dm["f10","tK"] <- 2
#
dm
```

```
    tA tB tC tD tE tF tG tH tI tK
f1   2  1  0  0  0  0  0  0  0  0
f2   1  2  1  0  0  0  0  0  0  0
f3   0  1  2  1  0  0  0  0  0  0
f4   0  0  1  2  1  0  0  0  0  0
f5   0  0  0  1  2  1  0  0  0  0
f6   0  0  0  0  1  2  1  0  0  0
f7   0  0  0  0  0  1  2  1  0  0
f8   0  0  0  0  0  0  1  2  1  0
f9   0  0  0  0  0  0  0  1  2  1
f10  0  0  0  0  0  0  0  0  1  2
```

Um etwas Übersicht zu behalten, haben wir im vorliegenden Fall die Listen grabweise eingegeben, d.h. sukzessive f1, f2, f3, … abgearbeitet und die Typen eingefüllt. Es geht naheliegenderweise auch umgekehrt, die Liste typweise zu organisieren. Da letzten Endes die einzelnen Zellen adressiert werden, ist auch ein chaotischen Vorgehen möglich. Ich empfehle aber, sich für einen Aufbau zu entscheiden, weil nur so eine gute Kontrolle des Eingegebenen möglich ist.

Apropos Kontrolle: Da große Tabellen unübersichtlich werden können, nachfolgend ein Code-Block, der für eine erste Datenkontrolle hilfreich sein kann:

```
ncol(dm)  # ruft die Anzahl der Spalten (Typen) ab
```

```
[1] 10
```

```
nrow(dm)  # ruft die Anzahl der Zeilen (Gräber, Fundekomplex) ab
```

```
[1] 10
```

```
#
colSums(dm) # ruft die Spaltensumme ab = Summe aller Typen-Anzahlen ab
```

```
tA tB tC tD tE tF tG tH tI tK
 3  4  4  4  4  4  4  4  4  3
```

```
rowSums(dm) # ruft die Zeilensummen ab = Summe der Typen in jedem Grab/Befundkomplex #
```

```
 f1  f2  f3  f4  f5  f6  f7  f8  f9 f10
  3   4   4   4   4   4   4   4   4   3
```

```
sum(colSums(dm))  # ruft die Gesamtzahl der eingegebenen Typen ab
```

```
[1] 38
```

Hinweis für die Dateneingabe: "Notepad++" ist ein sehr guter, kostenloser Editor, mit dem man solche - auch erheblich längeren - Datenlisten aufbauen und pflegen kann, die sich dann auch problemlos in **R** einlesen lässt. Quelle dort: https://notepad-plus-plus.org/downloads/ [6.1.2023].

Wir sind nun soweit, mit den listenweise eingegebenen Daten nun in **R** eine CA zu rechnen.[1] Damit es aber an der Eingabetabelle etwas zu sortieren gibt, was man hinterher auch erkennen kann (d. h. etwas Umsortierung erfolgen müsste), fügen wir unserem Datensatz nachträglich noch einen zeit-unspezifischen Typ "tU" hinzu (ähnlich wie **Abb. 11**). Das Anfügen eines zusätzlichen Grabes würde nach dem gleichen Prinzip erfolgen, wobei man allerdings "nrow" durch "ncol" ersetzen müsste und "cbind" durch "rbind".

```
# kreiere Vektor "tU" mit dem Wert "0" für so vielen Zeilen, wie "dm" Zeilen hat
tU <- rep(0, times=nrow(dm))
tU
```

```
 [1] 0 0 0 0 0 0 0 0 0 0
```

[1] Im Kap. 23 nutze ich z.T. Code von Georg Roth (FU Berlin): http://www.rchaeology.eu/ [6.1.2023].

```
#
# verbinde dm mit dem neuen Vektor tU, d.h. füge 1 neue Spalte an:
dm <- cbind(dm, tU)
dm
```

```
    tA tB tC tD tE tF tG tH tI tK tU
f1   2  1  0  0  0  0  0  0  0  0  0
f2   1  2  1  0  0  0  0  0  0  0  0
f3   0  1  2  1  0  0  0  0  0  0  0
f4   0  0  1  2  1  0  0  0  0  0  0
f5   0  0  0  1  2  1  0  0  0  0  0
f6   0  0  0  0  1  2  1  0  0  0  0
f7   0  0  0  0  0  1  2  1  0  0  0
f8   0  0  0  0  0  0  1  2  1  0  0
f9   0  0  0  0  0  0  0  1  2  1  0
f10  0  0  0  0  0  0  0  0  1  2  0
```

```
# und besetze einige Zellen mit "1":
dm["f2","tU"] <- 1
dm["f3","tU"] <- 1
dm["f4","tU"] <- 1
dm["f6","tU"] <- 1
dm["f7","tU"] <- 1
dm["f8","tU"] <- 1
dm
```

```
    tA tB tC tD tE tF tG tH tI tK tU
f1   2  1  0  0  0  0  0  0  0  0  0
f2   1  2  1  0  0  0  0  0  0  0  1
f3   0  1  2  1  0  0  0  0  0  0  1
f4   0  0  1  2  1  0  0  0  0  0  1
f5   0  0  0  1  2  1  0  0  0  0  0
f6   0  0  0  0  1  2  1  0  0  0  1
f7   0  0  0  0  0  1  2  1  0  0  1
f8   0  0  0  0  0  0  1  2  1  0  1
f9   0  0  0  0  0  0  0  1  2  1  0
f10  0  0  0  0  0  0  0  0  1  2  0
```

## 23 Von Fundlisten zur Matrix: der schnelle Weg

In Kap. 21 und 22 haben wir versucht, übliche Wege der Dateieingabe und -verwaltung in **R** näher zu verstehen und das Ganze "händisch" aufgebaut. Für ungeduldige Praktiker gibt es aber auch einen schnelleren, direkteren Weg zum Ziel, nämlich vor vorherein auf die listenartige Eingabe der Daten zu setzen und für die notwendige Umwandlung auf fertige R-Lösungen zu setzen. Genau das gehen wir in Kap. 23 in fünf Schritten durch.

### 23.1 Datensatz im Long-Format aus *xlsx-Tabelle einlesen.

Daten können in **R** im sog. *wide format* oder im sog. *long format* abgelegt werden. Das "Wide Format" entspricht unseren gewohnten Tabellen: jede Zeile ist ein Fall, jede Spalte ist eine Variable. Statt das "Long Format" lange zu erklären: die Tabelle "1c_ideal-matrix_longformat.xlsx" enthält den Inhalt unserer Übungsmatrix mit je 10 Typen und Gräbern im Long Format: jede nicht-leere Zelle ist ein Fall, d. h. eine (kurze) Datenzeile. Sie gibt an, in welchem Grab welcher Typ in welcher Häufigkeit vorkommt. Wir lesen diese Datei in **R** ein und schauen sie uns an:

```
library(readxl)
#
df <- read_excel("1c_ideal-matrix_longformat.xlsx")
df
```

```
# A tibble: 28 x 3
   Grab    Typ    Anzahl
   <chr>   <chr>   <dbl>
 1 grave-1 type-A      2
 2 grave-1 type-B      1
 3 grave-2 type-A      1
 4 grave-2 type-B      2
 5 grave-2 type-C      1
 6 grave-3 type-B      1
 7 grave-3 type-C      2
 8 grave-3 type-D      1
 9 grave-4 type-C      1
10 grave-4 type-D      2
# ... with 18 more rows
```

Man beachte: Weil ich vornan ein "Projekt" in **R** angelegt habe, "weiss" **R**, wo meine Datei liegt, d.h. ich muss keinen langen Pfad eingeben. Möglicherweise ist das bei Ihnen anders und Sie impotieren mit der Angabe des Pfads.

In der Beispieltabelle "1c_ideal-matrix_longformat.xlsx" erfolgt die Eingabe grabweise. Ein umgekehrtes Vorgehen in der Reihenfolge der Typen ist ebenso möglich. Bitte schauen Sie sich diese Tabelle sorgfältig an. Zunächst mag der Anblick ungewöhnlich sein. Das Befreit-Sein von den vielen Nullen in einer ev. großen Tabelle macht diese Art der Datenverwaltung nach kurzer Eingewöhnung sehr übersichtlich und vor allem weniger fehleranfällig als das chronische Zeilen- und Spalten-Halten-Müssen bei großen Tabellen (im Wide Format).

### 23.2 Prüfen von Voraussetzungen

In einer guten Kontingenztabelle ("Kombinationstabelle") kommt jeder Typ in mind. 2 Gräbern vor und jedes Grab enthält mindestens 2 unterschiedliche Typen. Nach der Dateneingabe sollte man dies prüfen. Bei unserem kleinen Bespiele mit je 10 Gräbern und Typen kann das anhand der Tabelle visuell geschehen, bei großen Tabellen setzen wir unterstützend **R** ein. Schritt 1: wie viele "Inzidenzen" gibt es, d.h. Zellen, die nicht leer sind und in denen eine Häufigkeit steht?

```
nrow(df)
```

```
[1] 28
```

Nun zählen wir, in wie vielen Gräbern jeder Typ vorkommt:

```
table(df$Typ)
```

```
type-A type-B type-C type-D type-E type-F type-G type-H type-I type-K
     2      3      3      3      3      3      3      3      3      2
```

```
#
# hist(table(df$Typ)) # z.B. so kann man das Ergebnis auch als Histogramm anschauen
```

Nun zählen wir, wie viele Typen in den Gräbern vorkommen:

```
table(df$Grab)
```

```
 grave-1 grave-10  grave-2  grave-3  grave-4  grave-5  grave-6  grave-7
       2        2        3        3        3        3        3        3
 grave-8  grave-9
       3        3
```

```
#
# hist(table(df$Grab)) # z.B. so kann man das Ergebnis auch als Histogramm anschauen
```

In allen Fällen sollte die Zahlen eine 2 oder höher zeigen. Wenn Sie ein "NA" (i.e. *not available;* fehlender Wert), eine Null oder eine 1 entdecken, bietet dieses Grab / dieser Typ keine Fundkombination und sollte aus dem Datensatz entfernt werden. Das kann manuell anhand der Eingabetabelle erfolgen oder später, wenn wir die Daten in die "wide form" übertragen haben. In frühen Stadien eines Projekts verbergen sich hinter den "NAs" und niedrigen Häufigkeiten gerne auch Tippfehler, weshalb zunächst Korrekturen direkt am Eingabedatensatz vermutlich effizienter sind.

Man beachte: nach der ersten Bereinigung dieser Art um seltene Gräber und Typen kann es vorkommen, dass dann erneut - dieses Mal andere - Gräber oder Typen zu selten sind, weil mit der ersten Bereinigung eine Kombination weggefallen sein kann.

Hinweis: Die Mindestzahl 2 begründet sich aus der Logik des Verfahren. Bei weniger als 2 liegt keine Fundkombination vor. Es kann allerdings durchaus erwogen werden, diesen Mindestwert nach archäologischen Überlegungen z. B. auch auf 3 oder 4 anzusetzen, um z. B. in einem ingesamt fundreichen Siedlungsmaterial "seltene Fälle" / Exotika von vornherein auszuschließen. Auch die andere Richtung des Themas Häufigkeit sollte durchdacht werden: Allzu häufige Typen sind möglicherweise zu unspezifisch, d. h. es könnte auch nach einer Obergrenze der Anzahl gefragt werden, um dann allzu häufige Typen von der weiteren Analyse auszuschließen werden.

### 23.3 Umwandeln vom "Long Format" ins "Wide Format":

```
library(tidyr)
library(tidyverse)
#
dfwide <- pivot_wider(data = df,
                      names_from = "Typ",
                      values_from = "Anzahl",
                      values_fill = 0)
dfwide
```

```
# A tibble: 10 x 11
   Grab  type-~1 type-~2 type-~3 type-~4 type-~5 type-~6 type-~7 type-~8 type-~9
   <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 grav~       2       1       0       0       0       0       0       0       0
 2 grav~       1       2       1       0       0       0       0       0       0
 3 grav~       0       1       2       1       0       0       0       0       0
```

```
 4 grav~       0       0       1       2       1       0       0       0       0
 5 grav~       0       0       0       1       2       1       0       0       0
 6 grav~       0       0       0       0       1       2       1       0       0
 7 grav~       0       0       0       0       0       1       2       1       0
 8 grav~       0       0       0       0       0       0       1       2       1
 9 grav~       0       0       0       0       0       0       0       1       2
10 grav~       0       0       0       0       0       0       0       0       1
# ... with 1 more variable: `type-K` <dbl>, and abbreviated variable names
#   1: `type-A`, 2: `type-B`, 3: `type-C`, 4: `type-D`, 5: `type-E`,
#   6: `type-F`, 7: `type-G`, 8: `type-H`, 9: `type-I`
```

Die Variable/Spalte “Grab” als Zeilenname deklarieren:Man beachte: nach der ersten Bereinigung dieser Art um seltene Gräber und Typen kann es vorkommen, dass dann erneut - dieses Mal andere - Gräber oder Typen zu selten sind, weil mit der ersten Bereinigung eine Kombination weggefallen sein kann.

Hinweis: Die Mindestzahl 2 begründet sich aus der Logik des Verfahren. Bei weniger als 2 liegt keine Fundkombination vor. Es kann allerdings durchaus erwogen werden, diesen Mindestwert nach archäologischen Überlegungen z. B. auch auf 3 oder 4 anzusetzen, um z. B. in einem ingesamt fundreichen Siedlungsmaterial “seltene Fälle” / Exotika von vornherein auszuschließen. Auch die andere Richtung des Themas Häufigkeit sollte durchdacht werden: Allzu häufige Typen sind möglicherweise zu unspezifisch, d. h. es könnte auch nach einer Obergrenze der Anzahl gefragt werden, um dann allzu häufige Typen von der weiteren Analyse auszuschließen werden.

Im Dataframe “dfwide” sind die Namen der Gräber/Fundkomplexe noch eine Datenspalte. Dies sieht zwar korrekt aus, ist jedoch für einige weitere Schritte der Verarbeitung (technisch) hinderlich. Daher speichern wir das Dataframe “dfwide” um zu “dfwide2” und erklären dort die Variable/Spalte “Grab” zum Zeilennamen:

```
print(dfwide)
```

```
# A tibble: 10 x 11
   Grab  type-~1 type-~2 type-~3 type-~4 type-~5 type-~6 type-~7 type-~8 type-~9
   <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 grav~       2       1       0       0       0       0       0       0       0
 2 grav~       1       2       1       0       0       0       0       0       0
 3 grav~       0       1       2       1       0       0       0       0       0
 4 grav~       0       0       1       2       1       0       0       0       0
 5 grav~       0       0       0       1       2       1       0       0       0
 6 grav~       0       0       0       0       1       2       1       0       0
 7 grav~       0       0       0       0       0       1       2       1       0
 8 grav~       0       0       0       0       0       0       1       2       1
```

```
 9 grav~       0       0       0       0       0       0       0       1       2
10 grav~       0       0       0       0       0       0       0       0       1
# ... with 1 more variable: `type-K` <dbl>, and abbreviated variable names
#   1: `type-A`, 2: `type-B`, 3: `type-C`, 4: `type-D`, 5: `type-E`,
#   6: `type-F`, 7: `type-G`, 8: `type-H`, 9: `type-I`
```

```
#
dfwide2 <- dfwide %>% remove_rownames %>% column_to_rownames(var="Grab")
print(dfwide2)
```

```
         type-A type-B type-C type-D type-E type-F type-G type-H type-I type-K
grave-1       2      1      0      0      0      0      0      0      0      0
grave-2       1      2      1      0      0      0      0      0      0      0
grave-3       0      1      2      1      0      0      0      0      0      0
grave-4       0      0      1      2      1      0      0      0      0      0
grave-5       0      0      0      1      2      1      0      0      0      0
grave-6       0      0      0      0      1      2      1      0      0      0
grave-7       0      0      0      0      0      1      2      1      0      0
grave-8       0      0      0      0      0      0      1      2      1      0
grave-9       0      0      0      0      0      0      0      1      2      1
grave-10      0      0      0      0      0      0      0      0      1      2
```

### 23.4 Sicherheit durch Überprüfen

Wir ermitteln die Anzahl der der Spalten/Typen und der Gräber/Zeilen:

```
ncol(dfwide[,-1]) # schließt die erste Spalte=Zeilenname aus, d.h. gibt die Anzahl der Typ
```

```
[1] 10
```

```
ncol(dfwide2)
```

```
[1] 10
```

```
#
nrow(dfwide)  # Anzahl der Gräber
```

```
[1] 10
```

```
nrow(dfwide2)
```

```
[1] 10
```

Beispiel: So könnte eine Spalte/ein Grab oder mehrere Gräber zugleich aus der Datentabelle komplett gelöscht werden. (Hier im Codeblock mit # als Kommentar markiert, d. h. deaktiviert.)

```
# dfwide$"type-A" <- NULL
# dfwide
```

### 23.5 Umformatieren des Dataframes in eine Matrix

Einige der R-Pakete, die wir im Folgenden für die CA verwenden, erwarten einen Dataframe als Eingabe, in unserem Fall also "dfwide2". Andere Pakete erwarten eine Matrix als Eingabe. Damit wir flexibel sind, verwandeln wir die Daten auch in eine Matrix, und zwar so, dass Spalte 1 = "Grab" als Zeilename der Matrix geführt wird:

```
matrix.please <- function(dfwide) {
    dm <- as.matrix(dfwide[,-1])
    rownames(dm) <- dfwide$Grab
    dm
}
#
dm <- matrix.please(dfwide)
dm
```

```
         type-A type-B type-C type-D type-E type-F type-G type-H type-I type-K
grave-1       2      1      0      0      0      0      0      0      0      0
grave-2       1      2      1      0      0      0      0      0      0      0
grave-3       0      1      2      1      0      0      0      0      0      0
grave-4       0      0      1      2      1      0      0      0      0      0
grave-5       0      0      0      1      2      1      0      0      0      0
grave-6       0      0      0      0      1      2      1      0      0      0
grave-7       0      0      0      0      0      1      2      1      0      0
grave-8       0      0      0      0      0      0      1      2      1      0
grave-9       0      0      0      0      0      0      0      1      2      1
grave-10      0      0      0      0      0      0      0      0      1      2
```

Fertig. Wir haben aus der Eingabe im Long Format eine Tabelle und eine Matrix gemacht, also den STand erreicht wie mit der Matrix "dm" am Ende von Kap. 22, mit der wir eine Korrespondenzanalyse (CA) rechnen können.

## 24 Durchführung einer CA mit R

Die Korrespondenzanalyse rechnen wir mit dem Paket "CA", bei dem u.a. Michael Greenacre Co-Autor ist (Nenadic & Greenacre, 2007). Es gehört nicht zu Basisaustattung von R, d. h. es muss zunächst installiert werden und dann mit "library(ca)" aktiviert werden. Sofern noch nicht geschehen, installieren Sie "ca" bitte wie oben bei "R-Pakete" beschrieben direkt von CRAN.

```
library(ca)
```

Das eigentliche Rechnen der CA braucht nur folgende zwei Anweisungen:

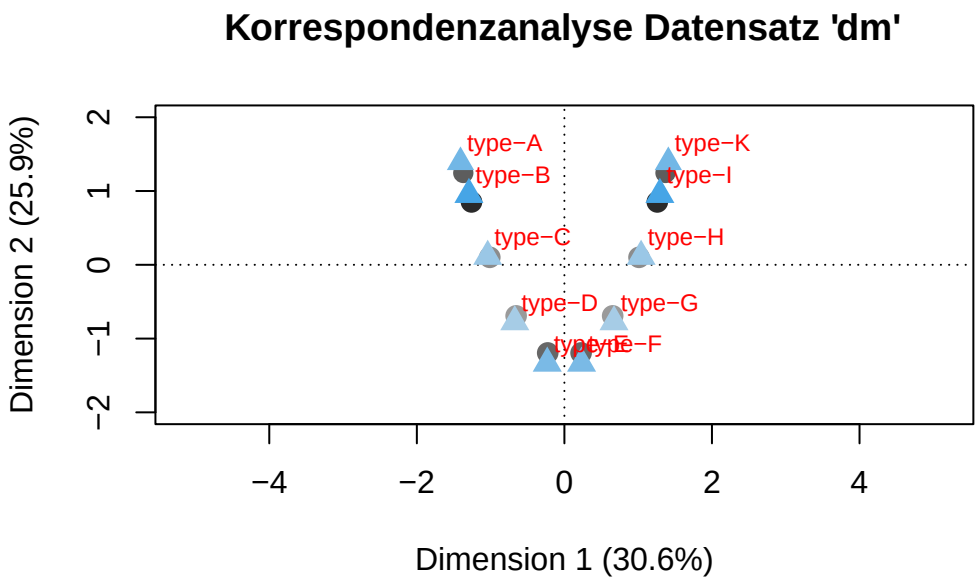
```
result <- ca(dm)
summary(result)
```

```
Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.946589  30.6  30.6  ********
 2      0.800791  25.9  56.4  ******
 3      0.600452  19.4  75.8  *****
 4      0.392855  12.7  88.5  ***
 5      0.218319   7.0  95.5  **
 6      0.098334   3.2  98.7  *
 7      0.032836   1.1  99.8
 8      0.006631   0.2 100.0
 9      0.000416   0.0 100.0
        -------- -----
 Total: 3.097222 100.0


Rows:
     name   mass  qlt  inr     k=1 cor ctr     k=2 cor ctr
1  | grv1 |   79  603  145 | -1370 330 157 |  1245 272 153 |
2  | grv2 |  105  837   94 | -1259 574 176 |   852 263  95 |
3  | grv3 |  105  403   87 | -1011 399 114 |   101   4   1 |
4  | grv4 |  105  354   87 |  -654 167  48 |  -692 187  63 |
```

```
5  | grv5 |  105  577   87 |  -226  20   6 | -1194 557 188 |
6  | grv6 |  105  577   87 |   226  20   6 | -1194 557 188 |
7  | grv7 |  105  354   87 |   654 167  48 |  -692 187  63 |
8  | grv8 |  105  403   87 |  1011 399 114 |   101   4   1 |
9  | grv9 |  105  837   94 |  1259 574 176 |   852 263  95 |
10 | gr10 |   79  603  145 |  1370 330 157 |  1245 272 153 |

Columns:
     name   mass  qlt  inr     k=1 cor ctr     k=2 cor ctr
1  | typA |   79  603  145 | -1370 330 157 |  1245 272 153 |
2  | typB |  105  837   94 | -1259 574 176 |   852 263  95 |
3  | typC |  105  403   87 | -1011 399 114 |   101   4   1 |
4  | typD |  105  354   87 |  -654 167  48 |  -692 187  63 |
5  | typE |  105  577   87 |  -226  20   6 | -1194 557 188 |
6  | typF |  105  577   87 |   226  20   6 | -1194 557 188 |
7  | typG |  105  354   87 |   654 167  48 |  -692 187  63 |
8  | typH |  105  403   87 |  1011 399 114 |   101   4   1 |
9  | typI |  105  837   94 |  1259 574 176 |   852 263  95 |
10 | typK |   79  603  145 |  1370 330 157 |  1245 272 153 |
```

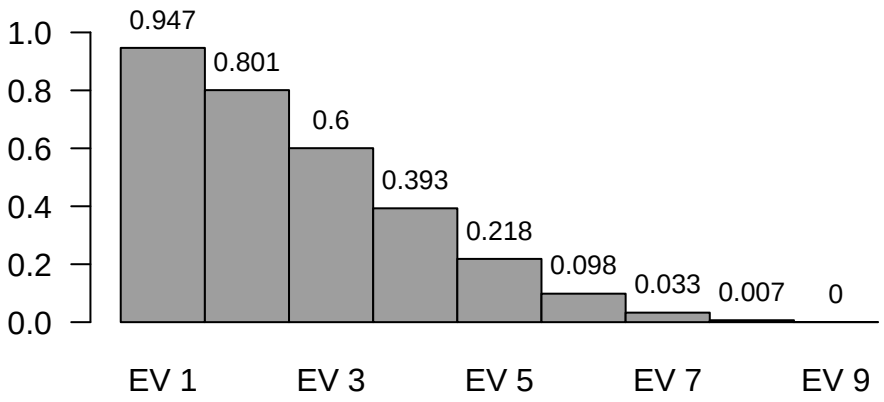
Statt die vielen Zahlen zu lesen, möchten wir erst einmal das übliche Streuungsdiagramm EV1 mit EV 2 anschauen:

```
plot(result, 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))
title(xlab="", ylab="", cex.lab=.9,
      main="Korrespondenzanalyse Datensatz 'dm'")
```




Nun lassen wir uns den Anteil der erklärten Inertia pro EV ausgeben:

```
Scree <- summary(result)[[1]][,2]
Name <- paste("EV", 1:length(Scree))
barplot(Scree, names.arg=Name,
        col=8, ylim=c(0,round(max(Scree)+max(Scree)/5,1)),
        space=0, las=1)
text((1:length(Scree))-.5, Scree, round(Scree,3), pos=3, cex=0.8)
title(main="Inertia pro Achse", font=2)
```




Anschließend betrachten wir die nach den Ergebnissen von EV1 neu geordnete Tabelle:

```
col_ord <- order(result$colcoord[,1])
row_ord <- order(result$rowcoord[,1])
#
ord_dm <- (dm[,1:10][row_ord,col_ord])
ord_dm
```

| | type-A | type-B | type-C | type-D | type-E | type-F | type-G | type-H | type-I | type-K |
|---|---|---|---|---|---|---|---|---|---|---|
| grave-1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-3 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| grave-4 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| grave-5 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 |
| grave-6 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| grave-7 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 |
| grave-8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 |
| grave-9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| grave-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |

Eine ästhetische Frage: mich stören die vielen Nullen beim schnellen Lesen der Tabelle, daher ersetze ich sie:

```
tU <- as.numeric(tU)
Tabelle <- as.data.frame(ord_dm)
Tabelle[Tabelle == 0] = "-"
Tabelle
```

```
         type-A type-B type-C type-D type-E type-F type-G type-H type-I type-K
grave-1       2      1      -      -      -      -      -      -      -      -
grave-2       1      2      1      -      -      -      -      -      -      -
grave-3       -      1      2      1      -      -      -      -      -      -
grave-4       -      -      1      2      1      -      -      -      -      -
grave-5       -      -      -      1      2      1      -      -      -      -
grave-6       -      -      -      -      1      2      1      -      -      -
grave-7       -      -      -      -      -      1      2      1      -      -
grave-8       -      -      -      -      -      -      1      2      1      -
grave-9       -      -      -      -      -      -      -      1      2      1
grave-10      -      -      -      -      -      -      -      -      1      2
```

Unser Beispiel ist bewusst klein gewählt. Wenn die Tabelle größer wird, möchte man sie so ausgeben, dass sie z. B. auf DIN A4-Seiten ausgedruckt werden kann. Wozu man sie in Streifen zerlegt. Das geht in unserem Beispiel z. B. so:

```
Tabelle[1:5]    # Ausgabe der ersten 5 Spalten
```

```
         type-A type-B type-C type-D type-E
grave-1       2      1      -      -      -
grave-2       1      2      1      -      -
grave-3       -      1      2      1      -
grave-4       -      -      1      2      1
grave-5       -      -      -      1      2
grave-6       -      -      -      -      1
grave-7       -      -      -      -      -
grave-8       -      -      -      -      -
grave-9       -      -      -      -      -
grave-10      -      -      -      -      -
```

```
Tabelle[,6:10]  # Ausgabe der nächsten 6 Spalten
```

```
         type-F type-G type-H type-I type-K
grave-1       -      -      -      -      -
```

```
grave-2       -      -      -      -      -
grave-3       -      -      -      -      -
grave-4       -      -      -      -      -
grave-5       1      -      -      -      -
grave-6       2      1      -      -      -
grave-7       1      2      1      -      -
grave-8       -      1      2      1      -
grave-9       -      -      1      2      1
grave-10      -      -      -      1      2
```

Möchte man diese Art von Tabellen in eine Textverarbeitung übergeben, muss man dort (für die Tabelle) eine Schrift wählen, bei der alle Zeichen die gleiche Breite aufweisen. Die typische Wahl ist Courier oder NewCourier. Eine gewöhnliche Textseite nimmt etwas mehr als 50 Zeichen auf, daher empfehle ich die Aufteilung der Tabelle in Portionen á 50 Spalten - was wir hier dank der Kleinheit der Übungstabelle nicht brauchen.

## 25 Die Kennzahlen

Kommen wir zurück auf die Kennzahlen, die wir oben zwar abgerufen, inhaltlich aber übersprungen hatten:

```
summary(result)
```

```
Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.946589  30.6  30.6  ********
 2      0.800791  25.9  56.4  ******
 3      0.600452  19.4  75.8  *****
 4      0.392855  12.7  88.5  ***
 5      0.218319   7.0  95.5  **
 6      0.098334   3.2  98.7  *
 7      0.032836   1.1  99.8
 8      0.006631   0.2 100.0
 9      0.000416   0.0 100.0
        -------- -----
 Total: 3.097222 100.0


Rows:
     name   mass  qlt  inr     k=1 cor ctr     k=2 cor ctr
```

```
1  | grv1 |   79  603  145 | -1370 330 157 |  1245 272 153 |
2  | grv2 |  105  837   94 | -1259 574 176 |   852 263  95 |
3  | grv3 |  105  403   87 | -1011 399 114 |   101   4   1 |
4  | grv4 |  105  354   87 |  -654 167  48 |  -692 187  63 |
5  | grv5 |  105  577   87 |  -226  20   6 | -1194 557 188 |
6  | grv6 |  105  577   87 |   226  20   6 | -1194 557 188 |
7  | grv7 |  105  354   87 |   654 167  48 |  -692 187  63 |
8  | grv8 |  105  403   87 |  1011 399 114 |   101   4   1 |
9  | grv9 |  105  837   94 |  1259 574 176 |   852 263  95 |
10 | gr10 |   79  603  145 |  1370 330 157 |  1245 272 153 |

Columns:
     name   mass  qlt  inr     k=1 cor ctr     k=2 cor ctr
1  | typA |   79  603  145 | -1370 330 157 |  1245 272 153 |
2  | typB |  105  837   94 | -1259 574 176 |   852 263  95 |
3  | typC |  105  403   87 | -1011 399 114 |   101   4   1 |
4  | typD |  105  354   87 |  -654 167  48 |  -692 187  63 |
5  | typE |  105  577   87 |  -226  20   6 | -1194 557 188 |
6  | typF |  105  577   87 |   226  20   6 | -1194 557 188 |
7  | typG |  105  354   87 |   654 167  48 |  -692 187  63 |
8  | typH |  105  403   87 |  1011 399 114 |   101   4   1 |
9  | typI |  105  837   94 |  1259 574 176 |   852 263  95 |
10 | typK |   79  603  145 |  1370 330 157 |  1245 272 153 |
```

Der oberste Block der Ausgabe informiert uns über das Gesamtergebnis. In unserem Übungsbeispiel werden 9 “dim” berechnet, d. h. neun Eigenvektoren. Wichtig ist die Spalte “%”, sie gibt den Anteil der Inertia an, den diese Spalte zur Gesamt-Inertia beiträgt. Daran sehen wir, dass die beiden ersten dim / EVs zusammen mehr als 55 % der Gesamt-Inertia (im mathematischen Sinne) “erklären”. Das ist ein hoher und guter Anteil. Zudem sehen wir, dass die %-Werte der folgenden dim / EVs deutlich tiefer liegen - auch das ein gutes Bild.

Bei den beiden folgenden Blöcken zu den Zeilen (*rows*) und Spalten (*columns*) sind die statistischen Kennzahlen zwecks besserer Lesbarkeit mit 1000 multipliziert. Die Kennzahl “mass” steht für das relative Gewicht / die Bedeutung der jeweiligen Zeile/Spalte. “qlt” ist die Qualität, “inr” die Inertia (angegeben in millionstel Prozent an der Gesamtinertia). Es folgen als “k=1” die erste Lösung (EV1) und als “k=2” die zweite Lösung (EV2) der CA: die Lage (Koordinate) im Raum EV1 und EV2, dann “cor” = der relative Beitrag dieser Zeile/Spalte zur Gesamtinertia und “ctr” = der absolute Anteil dieser Zeit/Spalte zur Tabellen-Interia der gesamten Zeile/Spalte. Nähere Erläuterungen bei Greenacre 2007.

Wichtig davon sind m. E. die relative Inertia “inr”. In unserem Beispiel sehen wir, dass die Ecken (f1 u. f2, sowie f9 u. f10) wichtig sind, d.h. stark zur Gesamlösung beitragen, während der tU, der “Typ unspezifisch” für die Gesamtlösung am wenigsten Bedeutung hat. Ansonsten

sind eher relevant die Koordinaten, welche die Spalten/Zeilen resp. Typen/Fundkomplexe im Raum der Eigenvektoren innerhaben. Sie werden wie folgt ausgelesen, wobei ich die Ausgabe mit [,1:3] auf die ersten drei dim / EVs eingegrenzt habe:

```
result$colcoord[,1:3]
```

```
             Dim1       Dim2       Dim3
type-A -1.4085522  1.3907628 -1.3582553
type-B -1.2941554  0.9521265 -0.4409776
type-C -1.0396173  0.1130987  0.8733770
type-D -0.6725014 -0.7734897  1.4013019
type-E -0.2325616 -1.3348076  0.6674287
type-F  0.2325616 -1.3348076 -0.6674287
type-G  0.6725014 -0.7734897 -1.4013019
type-H  1.0396173  0.1130987 -0.8733770
type-I  1.2941554  0.9521265  0.4409776
type-K  1.4085522  1.3907628  1.3582553
```

```
#
result$rowcoord[,1:3]
```

```
              Dim1       Dim2       Dim3
grave-1  -1.4085522  1.3907628 -1.3582553
grave-2  -1.2941554  0.9521265 -0.4409776
grave-3  -1.0396173  0.1130987  0.8733770
grave-4  -0.6725014 -0.7734897  1.4013019
grave-5  -0.2325616 -1.3348076  0.6674287
grave-6   0.2325616 -1.3348076 -0.6674287
grave-7   0.6725014 -0.7734897 -1.4013019
grave-8   1.0396173  0.1130987 -0.8733770
grave-9   1.2941554  0.9521265  0.4409776
grave-10  1.4085522  1.3907628  1.3582553
```

Die Abstände zwischen den EVs sind zwar keine absoluten Zahlen wie eine Zentimeterskala, aber sie dürfen interpretiert werden, als mehr oder weniger Nähe der Typen resp. Gräber zueinander.

Archäologische Einordnung: Bei großen Tabellen - und von denen ist hier ja die Rede - kann es geschehen, dass man manchmal nicht mehr jeden Typ (Spalte), jeden Fundkomplex (Zeile) oder jede Fundkombination im Auge hat und gut erinnert; in großen Tabellen ist es schwer, den Überblick zu wahren. Hier können die o.g. Kennzahlen hilfreich sein um Sinne “mit

dem Finger auf etwas Auffälliges zeigen". Einzelne Spalten oder Zeilen, die stark von den anderen Spalten / Zeilen abweichende Werte von "mass" oder "inr" zeigen, sind auffällig. Hier könnten (insbes. am Anfang einer Analyse) z. B. noch Eingabefehler vorliegen, oder eben Typen resp. Fundkomplexe sichtbar werden, die tatsächlich sehr ungewöhnlich sind. Aber ungewöhnliche Typen oder Fundkomplexe sind nicht per se "falsch", und warum sollte ein z. B. sehr fundreiches und zeittypisches Grab nicht einen hohen Einfluss auf eine gute Ordnung haben? Meines Erachtens ist am Ende die Archäologie und das Urteil des Bearbeiters entscheidend, und es wird mehr die Tabelle sein als die Kennzahlen, anhand derer man als versierte Archäologin etwas sieht und Entscheidungen trifft. Zum Beispiel die Entscheidung, einen Typ, einen Fundkomplex als unpassend aus der weiteren Analyse herauszunehmen.

## 26 Die geordnete Tabelle ausgeben

Es gibt verschiedene Wege, die resultierende Tabelle auszugeben. Alternativ wäre es auch möglich, die Tabelle innerhalb von **R** zunächst anzupassen und erst dann fertig auszugeben. Ich vermute jedoch, dass die Vorstellungen davon, wie die fertige Tabelle aussehen soll, sehr individuell sind und vor allem, dass Sie sich im Aufbereiten z. B. mit Hilfe Ihrer gewohnten Tabellenkalkulation (z. B. LO-Calc, MS-Excel) sicherer fühlen. Diese manuelle Nacharbeit kann z. B. darin bestehen, sich die Spaltenköpfe und Zeilennamen auch rechts und unten ans Ende der Tabelle zu setzen; die Nullen in den Zellen durch ein weniger sichtbares Zeichen wie z. B. "-" zu ersetzen; sich z. B. alle 10 Zeilen und/oder 10 Spalten einen Trenner in die Tabelle zu geben oder, eleganter, in der Tabellenkalkulation die Zeilen- und Spaltenrahmen zu entfernen und nur alle 10 Spalten und Zeilen einen solchen kräftigen Rahmen einzuziehen. So lange die aktive, forschende Arbeit an der Tabelle und der CA noch andauert, wird man sich vermutlich dafür entscheiden, nur das Nötigste am Layout zu tun und erst die Druckfassung gründlicher zu gestalten.

Für eine Ausgabe als *.csv oder *.xlsx-Datei ist es zunächst sinnvoll, aus der Matrix wieder einen Dataframe zu machen, weil die Ausgabefunktionen einen Dataframe erwarten.

```
ord_df <- as.data.frame(ord_dm)
```

Für die Ausgabe als *.csv-Datei wählen wir - je nach Bedarf - die R-Funktionen "write.csv" oder "write.csv2". Die Funktion write.csv schreibt das "englische" CSV-Format, bei dem der Punkt als Dezimalzeichen dient und das Komma als Trennzeichen zwischen den Spalten. Die Funktion write.csv2 schreibt das "deutsche" CSV-Format mit dem Komma als Dezimalzeichen und dem Semikolon als Spalten-Trenner.

```
# write.csv(ord_df, "D:\\StatArch\\CA\\Tabelle-CA1.csv", row.names=TRUE)
#
write.csv2(ord_df, "D:\\StatArch\\CA\\Tabelle-CA1.csv", row.names=TRUE)
```

Da in unserer Tabelle nur ganze Zahlen vorkommen, macht die Frage des Dezimalzeichens - Punkt oder Komma - keinen Unterschied, aber beim Spaltentrenner machen Komma oder Semikolon je nach Weiterverarbeitungsprogramm einen Unterscheid. Das Code-Beispiel hier hat das "englische" Format de-aktiviert und das "deutsche Format" aktiviert.

Wollen wir die Tabelle gleich als *.xlsx-Datei ausgeben, d. h. zur Weiterbearbeitung mit LO-Calc oder MS-Excel, ist dies z. B. mit Hilfe des Pakets "openxlsx" möglich. Bitte das Paket vor dem ersten Durchlauf von CRAN installieren.

```
library (openxlsx)
#
write.xlsx(ord_df, file="Tabelle-CA1.xlsx", overwrite=TRUE, asTable=TRUE,
           sheetName="CA1", tabColour="steelblue", rowNames=TRUE,
           startCol="A", startRow = 1)
```

Für alle Exporte gilt: wenn man wie oben empfohlen in RStudio ein Projekt angelegt hat, landen alle Ausgaben im Projektordner und werden im RStudio (Fenster unten rechts) auch unter "Files" angezeigt.

**Baustelle**: Ausgabe via RMarkdown/Quarto-Datei direkt als MS-Word-Datei? - z. B. mit "kable", siehe z. B.: https://r-empirische-wissenschaften.de/buch/ergebnisse-exportieren.html oder https://rlab.blogs.uni-hamburg.de/dig-skripte/Export_von_Daten_aus_R/index.html?s=Export%20von%20Daten%20aus%20R%20im%20Text-Format

# 27 Weitergehendes

## 27.1 Paket "CAinterprTools" von Gianmarco Alberti

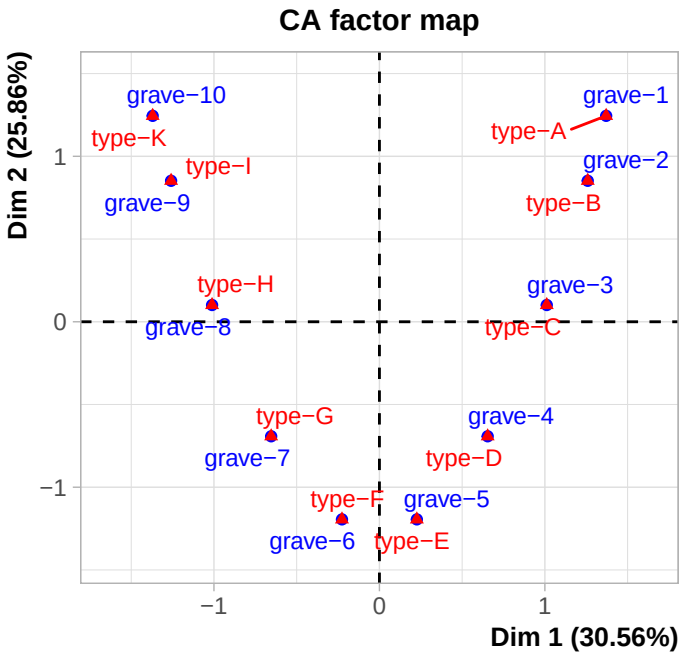
Das Paket "CAinterprTools" von Gianmarco Alberti (2015) möchte die Ergebnisse einer CA tiefer analysieren und bietet dazu verschiedene Werkzeuge an. Es nutzt zur Berechnung des CA das Paket "ca", d.h. die Ergebnisse im Sinne "die Ordnung der Zeilen und Spalten" ist mit dem Ergebnis mit dem Paket "ca" identisch. Daher kann die Ausgabe der neu geordneten Tabelle, die wie anhand der Ergebnisse von "ca" erzeugt hatten, verwendet werden. Das Paket CAinterprTools benötigt zur vollen Funktionsfähigkeit wiederum die Installation von zwei Pakete von CRAN aus: "devtools" und "CAinterprTools". Wie üblich: nur beim ersten Mal nötig, danach reicht das Aktivieren mit "library()". Also:

```
library(devtools)
library(CAinterprTools)
```

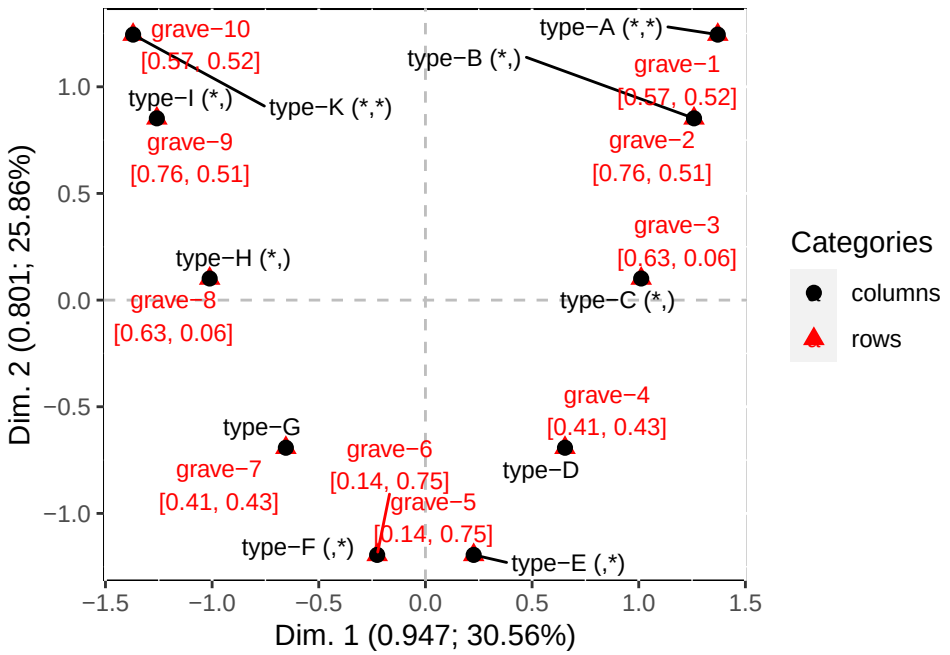
Neben dem üblichen Dokument zum Paket CAinterprTools auf CRAN bietet folgende Website eine gute Beschreibung von CAinterprTools: https://www.quantargo.com/help/r/latest/packages/CAinterprTools/1.1.0 [6.1.2023].

Das Paket CAinterprTools erwartet einen Dataframe als Eingabe, also arbeiten wir mit unserem Datensatz "dfwide2". Der folgende Code-Block demonstriert einige Beispiele aus diesem Paket, zu weiteren Möglichkeiten konsultiere man die Originaldokumentation von Alberti. Die Möglichkeiten dieses Pakets schätze ich sehr, sie fokussieren allerdings stark auf eine statistisch-numerische Analyse der Ergebnisse der Korrespondenzanalyse. Diese Analysen können - gerade bei großen Tabellen - wertvolle Hinweise zum weiteren Arbeiten geben wie z. B. Schwächen anzeigen, ersetzen jedoch die archäologische Analyse nicht. Der entscheidende Faktor jeder CA sind m. E. die eingegebenen Daten und letztlich die archäologische Arbeit an der Tabelle. Es gilt "garbarge in = garbage out", auch wenn die reine Statistik einer CA gute Ergebnisse anzeigt. Daher ist mir persönlich das Vorhandensein und die Anwendung einer archäologischen Prüfhypothese wichtiger als die mit dem Paket CAinterprTools verfolgte rein statistische Sicht. Gleichwie: die im folgenden Codeblock ausgwählten Beispiele zeigen einige Möglichkeiten des Pakets. Bei den beiden ersten Routinen lohnt es, anhand der Dokumentation des Pakets die verschiedenen Plot-Optionen für die Parabel durchzuprobieren.

```
caScatter(data=dfwide2, x=1, y=2, type=1)
```




```
#
caPlot(data=dfwide2, x=1, y=2)
```

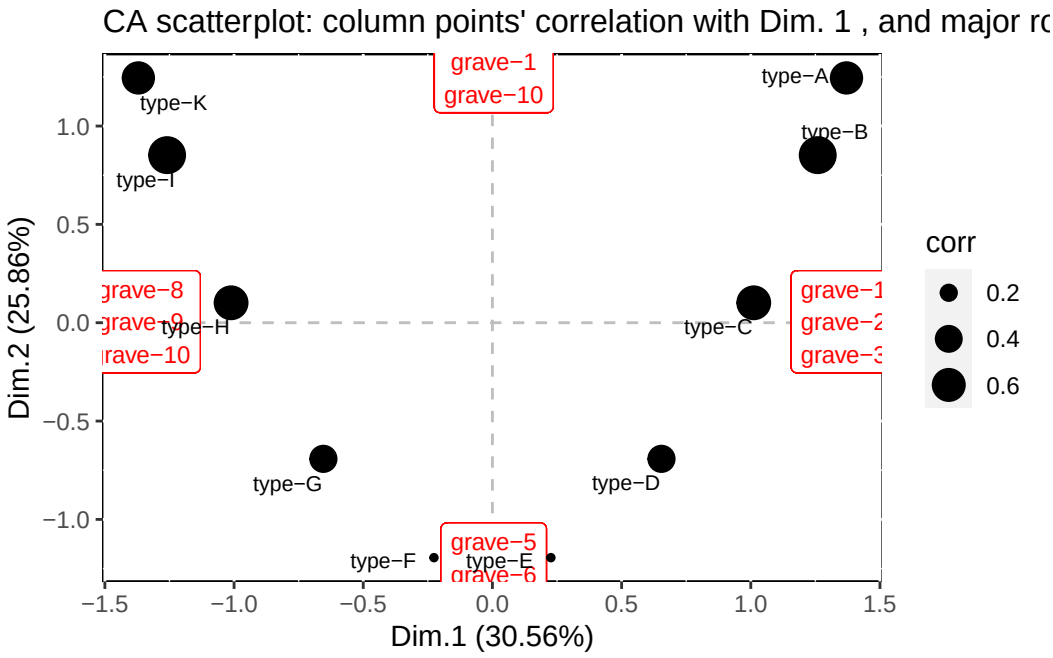



| | Categories | coord.1Dim | coord.2Dim | cntr.1Dim | cntr.2Dim | qlt.1Dim |
|---|---|---|---|---|---|---|
| grave-1 | rows | 1.3704200 | 1.2445507 | 15.663311 | 15.2701674 | 0.3303412 |
| grave-2 | rows | 1.2591201 | 0.8520287 | 17.629876 | 9.5425784 | 0.5743276 |
| grave-3 | rows | 1.0114729 | 0.1012086 | 11.376887 | 0.1346455 | 0.3992497 |
| grave-4 | rows | 0.6542954 | -0.6921721 | 4.760611 | 6.2977513 | 0.1670644 |
| grave-5 | rows | 0.2262658 | -1.1944782 | 0.569315 | 18.7548574 | 0.0199790 |
| grave-6 | rows | -0.2262658 | -1.1944782 | 0.569315 | 18.7548574 | 0.0199790 |
| grave-7 | rows | -0.6542954 | -0.6921721 | 4.760611 | 6.2977513 | 0.1670644 |
| grave-8 | rows | -1.0114729 | 0.1012086 | 11.376887 | 0.1346455 | 0.3992497 |
| grave-9 | rows | -1.2591201 | 0.8520287 | 17.629876 | 9.5425784 | 0.5743276 |
| grave-10 | rows | -1.3704200 | 1.2445507 | 15.663311 | 15.2701674 | 0.3303412 |
| type-A | columns | 1.3704200 | 1.2445507 | 15.663311 | 15.2701674 | 0.3303412 |
| type-B | columns | 1.2591201 | 0.8520287 | 17.629876 | 9.5425784 | 0.5743276 |
| type-C | columns | 1.0114729 | 0.1012086 | 11.376887 | 0.1346455 | 0.3992497 |
| type-D | columns | 0.6542954 | -0.6921721 | 4.760611 | 6.2977513 | 0.1670644 |
| type-E | columns | 0.2262658 | -1.1944782 | 0.569315 | 18.7548574 | 0.0199790 |
| type-F | columns | -0.2262658 | -1.1944782 | 0.569315 | 18.7548574 | 0.0199790 |
| type-G | columns | -0.6542954 | -0.6921721 | 4.760611 | 6.2977513 | 0.1670644 |
| type-H | columns | -1.0114729 | 0.1012086 | 11.376887 | 0.1346455 | 0.3992497 |
| type-I | columns | -1.2591201 | 0.8520287 | 17.629876 | 9.5425784 | 0.5743276 |
| type-K | columns | -1.3704200 | 1.2445507 | 15.663311 | 15.2701674 | 0.3303412 |

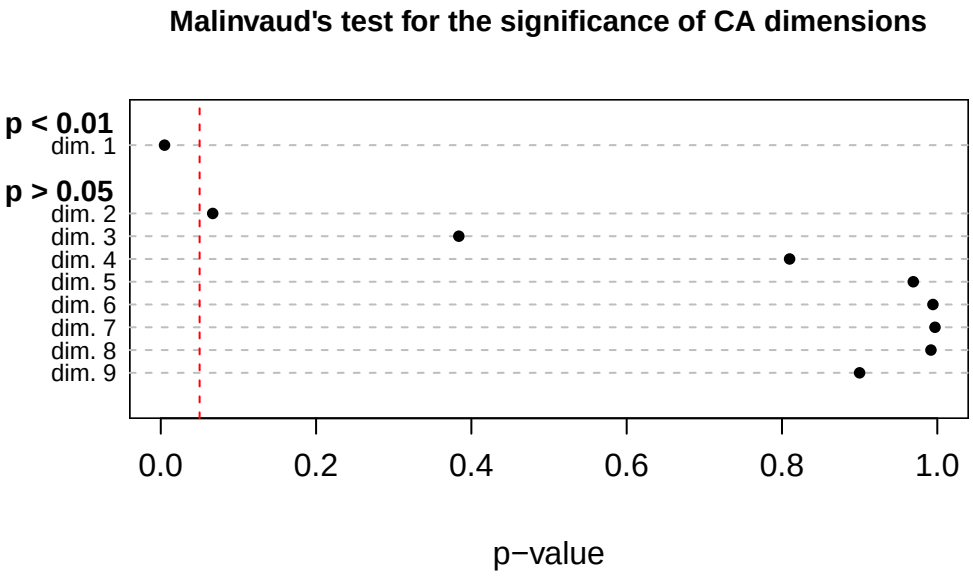
| | qlt.2Dim | corr.1Dim | corr.2Dim | majorcntr.1Dim | majorcntr.2Dim |
|---|---|---|---|---|---|
| grave-1 | 0.272446095 | 0.5747532 | 0.52196369 | | |

```
grave-2  0.262986686 0.7578440 0.51282228
grave-3  0.003997336 0.6318621 0.06322449
grave-4  0.186966717 0.4087351 0.43239648
grave-5  0.556791452 0.1413471 0.74618460
grave-6  0.556791452 0.1413471 0.74618460
grave-7  0.186966717 0.4087351 0.43239648
grave-8  0.003997336 0.6318621 0.06322449
grave-9  0.262986686 0.7578440 0.51282228
grave-10 0.272446095 0.5747532 0.52196369
type-A   0.272446095 0.5747532 0.52196369                *                *
type-B   0.262986686 0.7578440 0.51282228                *
type-C   0.003997336 0.6318621 0.06322449                *
type-D   0.186966717 0.4087351 0.43239648
type-E   0.556791452 0.1413471 0.74618460                                 *
type-F   0.556791452 0.1413471 0.74618460                                 *
type-G   0.186966717 0.4087351 0.43239648
type-H   0.003997336 0.6318621 0.06322449                *
type-I   0.262986686 0.7578440 0.51282228                *
type-K   0.272446095 0.5747532 0.52196369                *                *
```

```
#
caPercept(data=dfwide2)
```

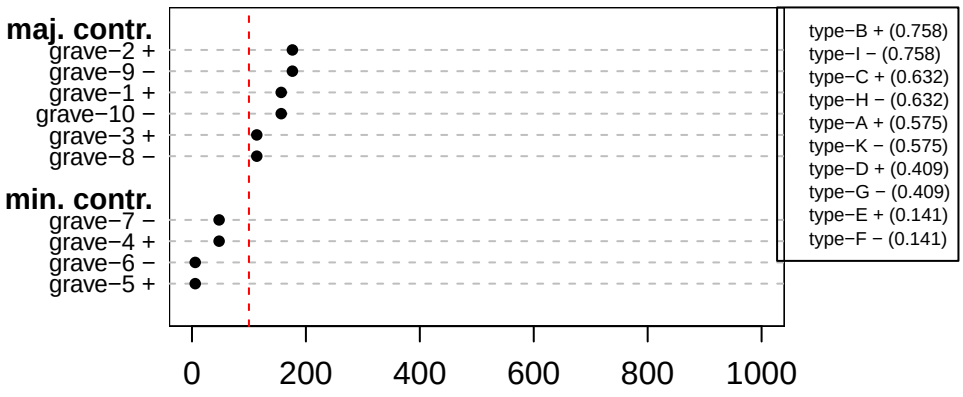


```
#
malinvaud(data=dfwide2)
```



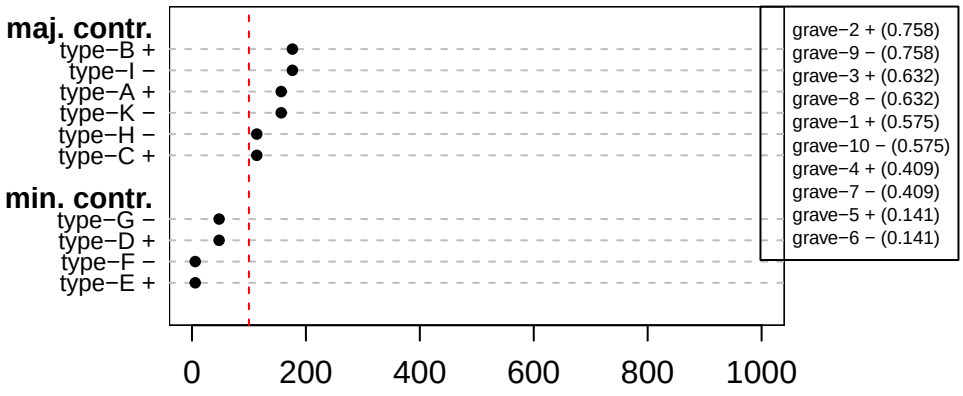

```
  K Dimension   Eigenvalue   Chi-square df     p-value  p-class
1 0    dim. 1 0.9465889754 117.69444444 81 0.004850732 p < 0.01
2 1    dim. 2 0.8007907766  81.72406338 64 0.066892636 p > 0.05
3 2    dim. 3 0.6004517976  51.29401387 49 0.383859315 p > 0.05
4 3    dim. 4 0.3928549668  28.47684556 36 0.809777888 p > 0.05
5 4    dim. 5 0.2183186442  13.54835682 25 0.969045816 p > 0.05
6 5    dim. 6 0.0983339213   5.25224834 16 0.994350125 p > 0.05
7 6    dim. 7 0.0328355896   1.51555933  9 0.997029687 p > 0.05
8 7    dim. 8 0.0066314464   0.26780693  4 0.991796443 p > 0.05
9 8    dim. 9 0.0004161044   0.01581197  1 0.899933267 p > 0.05
```

```
#
rows.cntr(data=dfwide2)
```



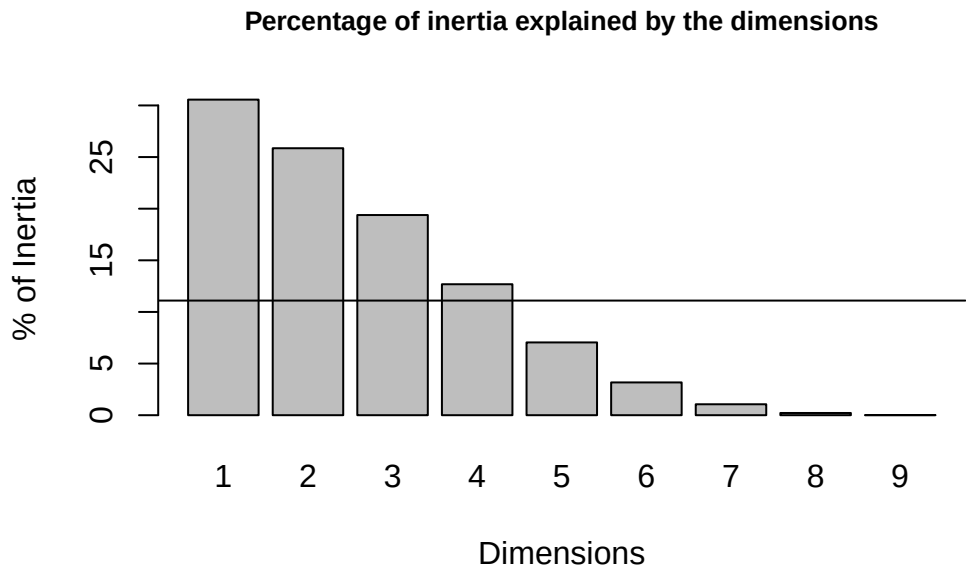

Row categories' contribution to Dim. 1 (in permills)

```
#
cols.cntr(data=dfwide2)
```




Column categories' contribution to Dim. 1 (in permills)

```
#
aver.rule(dfwide2)
```




eference line: threshold of an optimal dimensionality of the solution, according to the aver

Für die Details, die Bedeutung der einzelnen Ausgaben und die weiteren Möglichkeiten konsultiere man die Dokumentation zum gen. Paket. Wirklich nützlich ist m. E. der Malinvaud-Test, der einen p-Wert für die Signifikanz der Eigenvektoren (Dimensionen) ausgibt. In unserem Fall zeigen die beiden ersten EVs einen niedrigen P-Wert, der danach sprunghaft ansteigt. Nur der erste EV zeigt einen P-Wert unter 0.05, d.h. ist signifikant. Das sagt nicht unbedingt etwas über die archäologische Brauchbarkeit der ersten oder der ersten beiden Achsen, enthält aber die klare Aussage, dass wir die Achsen EV3 ff. im Grunde gar nicht anschauen brauchen.

Bei wirklich großen Tabellen können die beiden Folgegrafiken, die die Bedeutung der einzelnen Gräber resp. Typen für das Gesamtergebnis anzeigen, als Hinweisgeber sehr nützlich sein und ev. kritische Gräber / Typen anzeigen.

### 27.2 CA mit Bootstrapping

"Bootstrapping" (d.h. sich an den Schnürsenkeln aus dem Sumpf ziehen) bedeutet in der Statistik: systematisch aus einem Datensatz je einen oder mehrere Fälle herausnehmen, die Berechnungen durchführen, den herausgenommenen Fall in den Datensatz zurücklegen, den Prozess anschließend unter Herausnahme eines anderen Falls wiederholen, usw., usf.

Warum das Ganze? Es geht um die Sicherheit und Unsicherheit der geschätzten Parameter. In der parametrischen Statistik macht man Annahmen über die "Verteilung" der Daten. Man geht für Messwerte z. B. von einer Normalverteilung aus (und prüft per Test, ob diese Annahme zutrifft) und kann unter dieser Annahme die übliche Schwankung eines Parameters angeben. Beispielsweise die Standardabweichung für den Mittelwert - eine Spanne, in der ca. 2/3 aller Fälle liegen. Zudem lässt sich ein SEE "standard error of estimation" berechnen, der die Güte der Parameterschätzung angibt., also z. B. des Mittelwerts selbst.

Bei einer Korrespondenzanalyse ist dieser Weg der parametrischen Statistik nicht möglich: Häufigkeiten sind eine andere Art vion Information, sie unterliegen meist keiner Normalverteilung. Daher gibt es für die Lage der Punkte im Raum der Eigenvektoren keine Entsprechung zum SEE und zur Standardabweichung. Eine Möglichkeit, dieses Defizit zu umgehen und eine Kennzahl für die statistische Stabilität der Ergebnissen zu gewinnen, ist das Bootstrapping. Mit ihm lassen sich Wertebereiche angeben, innerhalb derer für jeden Typ und für jedes Grab / Fundkontext die Werte seiner Eigenvektoren schwanken.

Zur Umsetzung benutzen wir das R-Paket "cabootcrs" (Ringrose 2022), das man von CRAN installieren kann. Die ausführliche Dokumentation findet man auch dort: https://www.rdocumentation.org/packages/cabootcrs/versions/2.1.0/topics/cabootcrs [6.1.2023] und dort: https://github.com/cran/cabootcrs [6.1.2023].

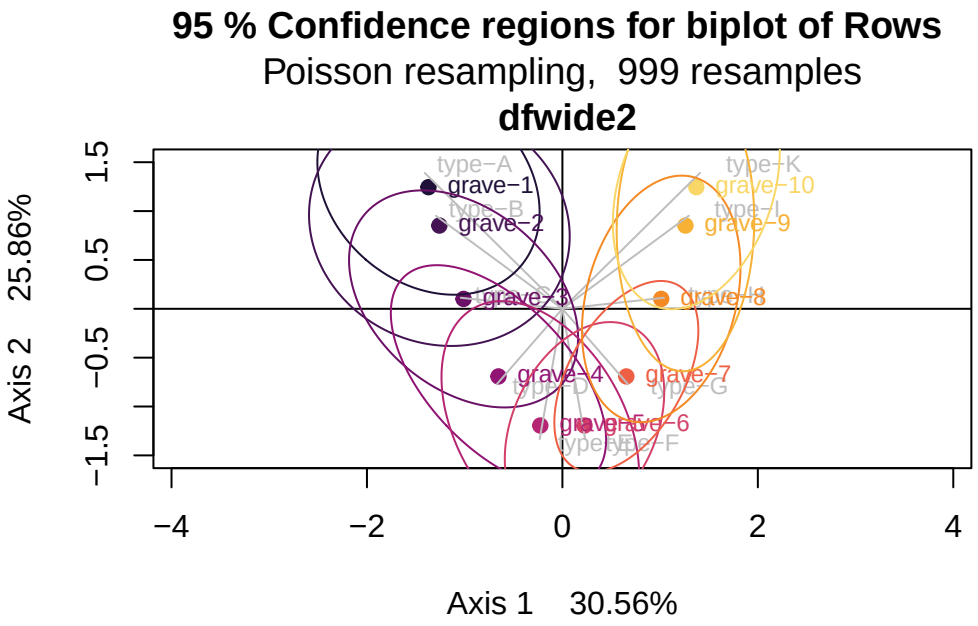
```
library(cabootcrs)
```

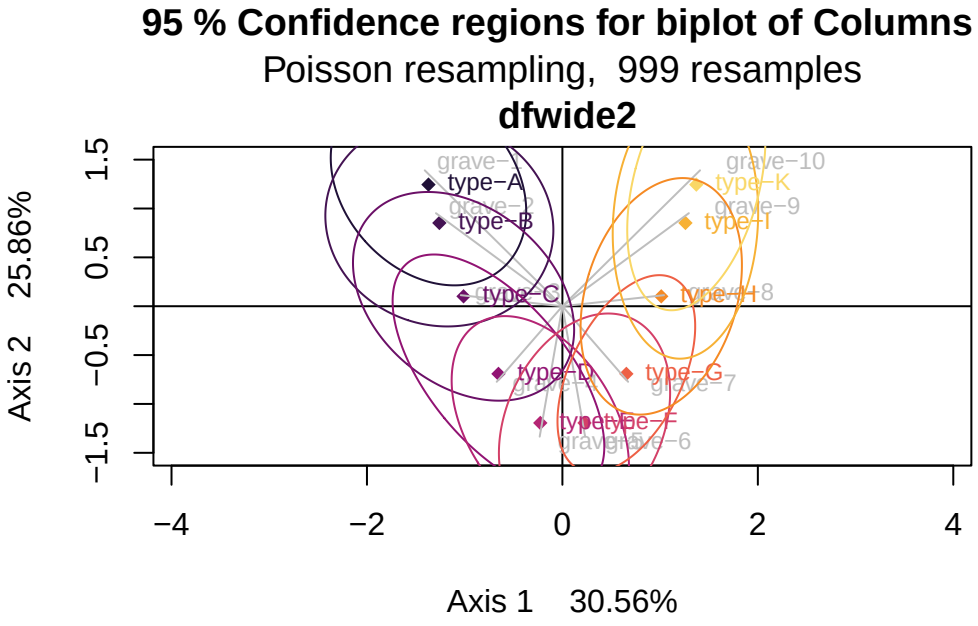
Um einen ersten Eindruck zu gewinnen, führen wir das Programm mit unseren Daten ohne weitere Parameter erst einmal aus. Man beachte: bei jedem Schritt des Bootstrappings wird eine CA gerechnet und deren Kernergebnisse gespeichert! Als Mindestmenge gelten 999 Interationen (was auch der Default bei cabootcrs ist). Ist ein CA-Projekt reif, d.h. weit gediehen und (fast) ausgearbeitet, wird man diesen Wert auf z. B. 2.000 oder 5.000 Iterationen hochsetzen. Kurz: hier ist Rechenpower des PCs und etwas Geduld gefragt, denn schon 999 CAs sind eine nenneswerte Menge.

```
results <- cabootcrs(dfwide2)
```

```
    SUMMARY RESULTS for Correspondence Analysis: dfwide2

Poisson resampling

Total inertia  3.097222

Inertias, percent inertias and cumulative percent inertias

    Inertia   %   Cum. %
1 0.9465890 30.56  30.56
```

```
2 0.8007908 25.86  56.42
3 0.6004518 19.39  75.80
4 0.3928550 12.68  88.49
5 0.2183186  7.05  95.54
6 0.0983339  3.17  98.71
7 0.0328356  1.06  99.77
8 0.0066314  0.21  99.99
9 0.0004161  0.01 100.00

Principal coords, std devs; rep and ctr (per mil); mass (per mil); 2-d rep (per mil)

Rows :
           Axis 1 StDev Rep Ctr   Axis 2 StDev Rep Ctr   Mass Quality
grave-1    -1.370 0.351 330 157    1.245 0.340 272 153     79     603
grave-2    -1.259 0.427 574 176    0.852 0.395 263  95    105     837
grave-3    -1.011 0.424 399 114    0.101 0.402   4   1    105     403
grave-4    -0.654 0.364 167  48   -0.692 0.377 187  63    105     354
grave-5    -0.226 0.335  20   6   -1.194 0.422 557 188    105     577
grave-6     0.226 0.306  20   6   -1.194 0.397 557 188    105     577
grave-7     0.654 0.299 167  48   -0.692 0.396 187  63    105     354
grave-8     1.011 0.303 399 114    0.101 0.471   4   1    105     403
grave-9     1.259 0.225 574 176    0.852 0.477 263  95    105     837
grave-10    1.370 0.237 330 157    1.245 0.346 272 153     79     603

Columns :
         Axis 1 StDev Rep Ctr   Axis 2 StDev Rep Ctr   Mass Quality
type-A   -1.370 0.306 330 157    1.245 0.315 272 153     79     603
type-B   -1.259 0.434 574 176    0.852 0.394 263  95    105     837
type-C   -1.011 0.416 399 114    0.101 0.392   4   1    105     403
type-D   -0.654 0.372 167  48   -0.692 0.420 187  63    105     354
type-E   -0.226 0.328  20   6   -1.194 0.395 557 188    105     577
type-F    0.226 0.314  20   6   -1.194 0.401 557 188    105     577
type-G    0.654 0.300 167  48   -0.692 0.430 187  63    105     354
type-H    1.011 0.306 399 114    0.101 0.448   4   1    105     403
type-I    1.259 0.258 574 176    0.852 0.481 263  95    105     837
type-K    1.370 0.206 330 157    1.245 0.374 272 153     79     603
```



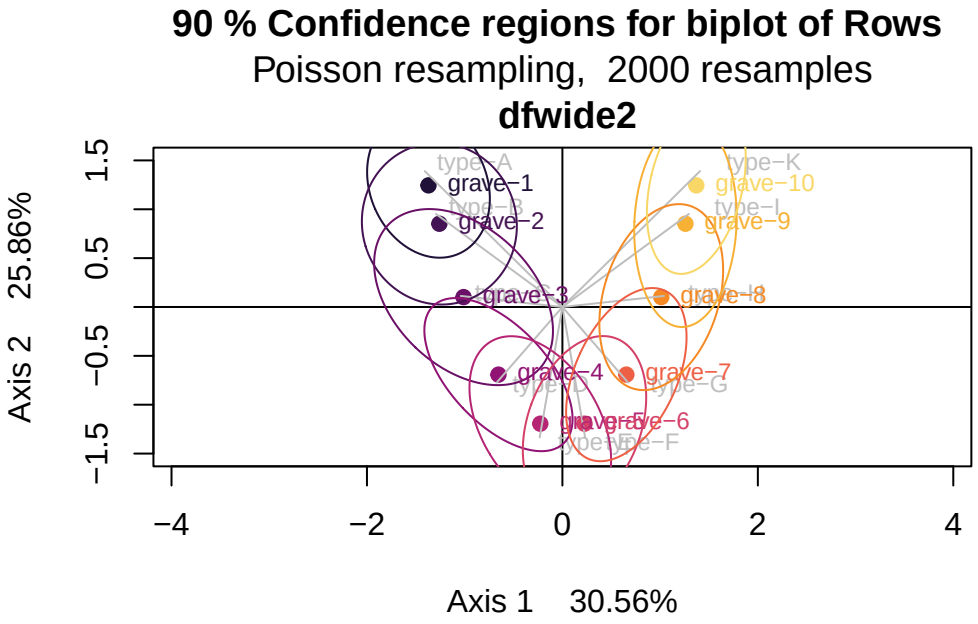

Nachfolgend bestellen wir 2000 Bootstrap-Stichproben und für die Plots ein etwas engeres Konfidenzintervall, dass “nur” 90% aller Fälle beinhaltet, d. h. an den Enden je 5 % der Fälle

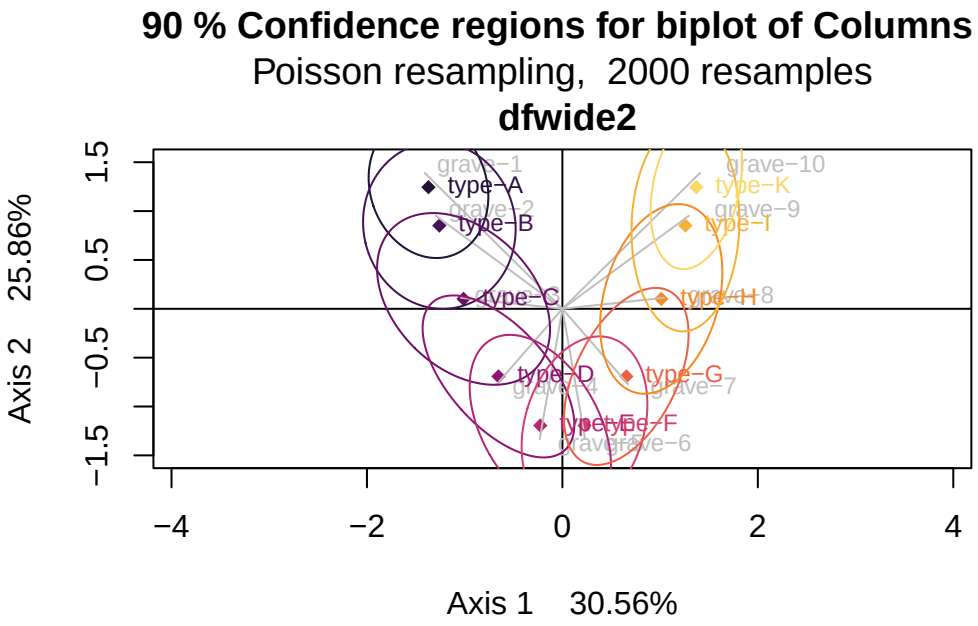
verwirft.

```
results <- cabootcrs(dfwide2, nboots=2000, crpercent=90)
```

```
    SUMMARY RESULTS for Correspondence Analysis: dfwide2

Poisson resampling

Total inertia  3.097222

Inertias, percent inertias and cumulative percent inertias

    Inertia   %   Cum. %
1 0.9465890 30.56  30.56
2 0.8007908 25.86  56.42
3 0.6004518 19.39  75.80
4 0.3928550 12.68  88.49
5 0.2183186  7.05  95.54
6 0.0983339  3.17  98.71
7 0.0328356  1.06  99.77
8 0.0066314  0.21  99.99
9 0.0004161  0.01 100.00

Principal coords, std devs; rep and ctr (per mil); mass (per mil); 2-d rep (per mil)

Rows :
           Axis 1 StDev Rep Ctr   Axis 2 StDev Rep Ctr   Mass Quality
grave-1    -1.370 0.297 330 157    1.245 0.350 272 153     79     603
grave-2    -1.259 0.399 574 176    0.852 0.416 263  95    105     837
grave-3    -1.011 0.417 399 114    0.101 0.410   4   1    105     403
grave-4    -0.654 0.385 167  48   -0.692 0.398 187  63    105     354
grave-5    -0.226 0.327  20   6   -1.194 0.404 557 188    105     577
grave-6     0.226 0.318  20   6   -1.194 0.452 557 188    105     577
grave-7     0.654 0.315 167  48   -0.692 0.453 187  63    105     354
grave-8     1.011 0.298 399 114    0.101 0.448   4   1    105     403
grave-9     1.259 0.241 574 176    0.852 0.485 263  95    105     837
grave-10    1.370 0.220 330 157    1.245 0.393 272 153     79     603

Columns :
         Axis 1 StDev Rep Ctr   Axis 2 StDev Rep Ctr   Mass Quality
type-A   -1.370 0.292 330 157    1.245 0.345 272 153     79     603
type-B   -1.259 0.394 574 176    0.852 0.430 263  95    105     837
```

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| type-C | -1.011 | 0.406 | 399 | 114 | 0.101 | 0.403 | 4 | 1 | 105 | 403 |
| type-D | -0.654 | 0.382 | 167 | 48 | -0.692 | 0.408 | 187 | 63 | 105 | 354 |
| type-E | -0.226 | 0.331 | 20 | 6 | -1.194 | 0.424 | 557 | 188 | 105 | 577 |
| type-F | 0.226 | 0.318 | 20 | 6 | -1.194 | 0.450 | 557 | 188 | 105 | 577 |
| type-G | 0.654 | 0.321 | 167 | 48 | -0.692 | 0.457 | 187 | 63 | 105 | 354 |
| type-H | 1.011 | 0.296 | 399 | 114 | 0.101 | 0.458 | 4 | 1 | 105 | 403 |
| type-I | 1.259 | 0.251 | 574 | 176 | 0.852 | 0.492 | 263 | 95 | 105 | 837 |
| type-K | 1.370 | 0.215 | 330 | 157 | 1.245 | 0.386 | 272 | 153 | 79 | 603 |






Lesehilfe: Im vorliegenden Fall sind die Ellipsen, d.h. die Konfidenzintervalle um den Schwerpunkt eines Grabs oder Typs (EV1/axis 1, EV2/axis 2) mehr ähnlich groß als unterschiedlich groß. Bei großen Tabellen wird man die sich überlagernden Ellipsen im Plot kaum mehr erkennen können. Kurz: die Plots sehen irgendwie gut aus, sind jedoch für das ernsthafte Arbeiten an großen Tabellen kaum geeignet. Daher wird man sich an den ausgegebenen Tabellen orientieren, genauer an der Standardabweichung ("StDev") der Eigenwerte auf den Achsen 1 und 2. In unserem kleinen Testfall sehen wir, dass diese mehr ähnlich denn stark unterschiedlich ausfallen und daher keine starke Aussage ergeben. Aber das dürfte bei größeren und echten Datensätzen anders ausfallen - wo dann Gräber / Typen mit geringer Standardabweichung sehr stabil positioniert sind und solche mit hoher Standardabweichung als weniger zuverlässig eingeordnet betrachtet werden können. Solche Instabilitäten können wiederum anzeigen, dass eine Fundkombination eher untypisch ist, dass der betreffende Typ zu unscharf definiert ist, usw.

### 27.3 Verbleibende Baustellen

Inhaltlich bin ich mit dieser Einführung noch nicht fertig, obwohl der Text im aktuellen Zustand für Sie schon den Start in das selbständige forschende Arbeiten mit Seriation und Korrespondenzanalyse ermöglicht. Meine wesentlichen Baustellen sind: das paket "anacor" durchleuchten für eine kanonische Korrespondenzanalyse; nach all den Übungsdateien einen "echten" Fall durchspielen.

# 28 Anhang / Apparat

## Abkürzungen

CA correspondence analysis, Korrespondenzanalyse

CCA canonical correspondence analysis, Kanonische Korrespondenzanalyse

DCA detrended correspondence analysis

PCA principle component analysis, Hauptkomponentenanalyse

## Literatur


Alberti, G. (2013). An R script to facilitate Correspondence Analysis. A guide to the use and the interpretation of results from an archaeological perspective. *Archeologia e Calcolatori, 24*, p. 25-53. - Online: http://www.progettocaere.rm.cnr.it/databasegestione/open_block_pages.asp?IDyear=2013-01-01 [6.1.2023].

Alberti, G. (2015). CAinterprTools: An R package to help interpreting Correspondence Analysis' results. *SoftwareX 5 (2015)*: http://dx.doi.org/10.1016/j.softx.2015.07.001 [6.1.2023].

Bayliss, A., Hines, J., Høilund Nielsen, K., McCormac, G. & Scull, Chr. (2013). *Anglo-Saxon graves and grave goods of the 6th and 7th centuries AD: a chronological framework.* Edited by J. Hines & A. Bayliss (The Society for Medieval Archaeology Monograph 33). London: The Society for Medieval Archaeology.

Benzécri, J.-P. (1976). L' analyse des données II: L' analyse des correspondances. París: Dunod.

Brongers, J. A. & Wijnman, H. F. (1968). Chronological classification of roemers with the help of 17th century paintings in the Low Countries. *Rotterdam Papers I,* p. 15-22.

Chernick, M. R. (1999). *Bootstrap Methods. A practitioner's guide.* (Wiley Series in probability and statistics). New York: John Wiley & Sons.

Christensen, C. M. (1997). *The innovator's dilemma: when new technologies cause great firms to fail.* Boston, Mass.: Harvard Business School Press.

Clarke, D. L. (1970). *Beaker pottery of Great Britain and Ireland.* Cambridge: University Press.

de Leeuw, J. (2013). *Correspondence analysis of archaeological abundance matrices.* Online: https://www.researchgate.net/publication/293114533_Correspondence_analysis_of_archaeological_abundance_matrices [6.1.2023].

Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* (Monographs on Statistics and Applied Probability, 57). New York: Chapmann & Hall.

Eggert, M. K. H., Kurz, S. & Wotzka, H.-P. (1980). Historische Realität und archäologische Datierung: Zur Aussagekraft der Kombinationsstatistik. *Praehistorische Zeitschrift, 55(1)*, p. 110-145.

Field, A., Miles, J. & Field, Z. (2012). *Discovering statistics using R.* London: Sage.

Ford, J. A. (1962). *A quantitative method for deriving cultural chronology* (Technical Manual 1). Washington, D.C.: Pan American Union.

Gatermann, H. (1942). *Die Becherkulturen der Rheinprovinz.* Würzburg: Triltsch. Open Access: https://digi.ub.uni-heidelberg.de/diglit/gatermann1942/0003/image,info

Gebühr, M. (1970). Beigabenvergesellschaftung in mecklenburgischen Gräberfeldern der älteren römischen Kaiserzeit. *Neue Ausgrabungen und Forschungen in Niedersachsen, 6*, p. 93-116.

Giesler, J. (1981). Untersuchungen zur Chronologie der Bijelo Brdo-Kultur. Ein Beitrag zur Archäologie des 10. und 11. Jahrhunderts im Karpatenbecken. *Praehistorische Zeitschrift, 56(1),* p. 3-221. DOI:10.1515/prhz.1981.56.1.3

Goldmann, K. (1972). Zwei Methoden chronologischer Gruppierung. *Acta Praehistorica et Archaeologica, 3,* p. 1-34. Open Access: https://journals.ub.uni-heidelberg.de/index.php/apa/article/view/67902

Good, Ph. I. (2013). *Introduction to statistics through resampling methods and R.* 2nd ed. Hoboken NY: Wiley.

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

Greenacre, M. J. (2007). *Correspondence analysis in practice.* 2nd ed. Boca Raton: Chapman & Hall.

Hammer, Ø., Harper, D. A. T. & Ryan, P. D. (2001). PAST: Paleontological statistics software package for education and data analysis. *Palaeontologia Electronica, 4(1)*: 9pp.

Hair, J. F., Black, W. C., Babin, B. J. & Anderson, R. E. (2010). *Multivariate data analysis.* 7th ed. Upper Saddle River: Prentice Hall.

Hartung, J. & Elpelt, B. (2007). *Multivariate Statistik. Lehr- und Handbuch der angewandten Statistik.* 7. Aufl. München: de Gruyter Oldenbourg.

Ihm, P. (1983). Korrespondenzanalyse und Seriation. *Archäologische Informationen, 6,* p. 8-21. Open Access: https://journals.ub.uni-heidelberg.de/index.php/arch-inf/article/view/27644

Ihm, P. (2005). A Contribution to the History of Seriation in Archaeology. In *Classification - the Ubiquitous Challenge. Proceedings of the 28th Annual Conference of the Gesellschaft für Klassifikation e.V. University of Dortmund, March 9-11, 2004* (Studies in Classification,

Data Analysis, and Knowledge Organization 2005). (p. 307-316). Berlin: Springer. DOI 10.1007/3-540-28084-7_34

Ihm, P. & van Groenewoud, H. (1984). Correspondence Analysis and Gaussian Ordination. *COMPSTAT Lectures, 3,* p. 5-60.

Kabacoff, R. I. (2022). *R in action.* 3rd edition. Shelter Island: Manning.

Kjeld Jensen, C. & Høilund Nielsen, K. (1997). Burial data and correspondence analysis. In Kjeld Jensen, C. & Høilund Nielsen, K. (eds). *Burial & society: the chronological and social analysis of archaeological burial data* (p. 29-61). Aarhus: Aarhus University Press. - Verfügbar z. B. bei Academia.edu: https://www.academia.edu/5050714/Burial_Data_and_CA [6.1.2023].

Koch, U. (1977). *Das Reihengräberfeld bei Schretzheim* (Germanische Denkmäler der Völkerwanderungszeit, A 13). Berlin: Gebr. Mann.

Koch, U. (2004). Schretzheim §2 Archäologisches. *Reallexikon der Germanischen Altertumskunde Bd. 27* (p. 294-302). Berlin: de Gruyter.

Legendre, P. & Gallagher, E. D. (2001). Ecologically meaningful transformations for ordination of species data. *Oecologia, 129*, p. 271-280. DOI: 10.1007/s004420100716

Lipo, C. P., Madsen, M. E. & Dunnell, R. C. (2015). A theoretically-sufficient and computationally-practical technique for deterministic frequency seriation. *PloS One, 10(4)*: e0124942 (online 29.4.2015). DOI: 10.1371/journal.pone.0124942

Madsen, T. (2007). *Mutivariate Data analysis with PCA, CA and MS* (ungedruckt). Online: http://archaeoinfo.dk/PDF%20files/2007%20Multivariate%20data%20analysis.pdf [6.1.2023].

Montelius, O. (1885). *Om tidsbestämning inom bronsåldern.* Stockholm: På Akademiens Förlag. Online: https://openlibrary.org/books/OL22888482M/Om_tidsbest%C3%A4mning_inom_brons%C3%A5ldern [6.1.2023].

Montelius, O. (1996). *Dating in the Bronze Age.* Stockholm: Kungl. Vitterhets Historie och Antikvitets akademien.

Montelius, O. (1903). *Die typologische Methode.* Stockholm: Selbstverlag des Verfassers.

Müller, J. & Zimmermann, A. (eds.) (1997). *Archäologie und Korrespondenzanalyse: Beispiele, Fragen, Perspektiven.* (Internationale Archäologie, 23). Espelkamp: Marie Leidorf.

Muenchen, R. A. (2022). The popularity of Data Science Software. Blog r4stats.com, 10.10.2022. https://r4stats.com/articles/popularity/ [6.1.2023].

Nenadic, O. & Greenacre, M. (2007). Correspondence Analysis in R, with two- and three-dimensional graphics: The ca package. *Journal of Statistical Software*, **20**(3), 1-13. Open Access:

Neuffer, E. M. (1965). Eine statistische Bearbeitung von Kollektivfunden. *Bonner Jahrbücher, 165*, p. 28-56. https://journals.ub.uni-heidelberg.de/index.php/bjb/article/view/73501

O'Brien, M. J., Lyman, R. L. & Darwent, J. (2000). Time, space and marker types: James A. Ford's 1936 chronology for the Lower Mississippi Valley. *Southeastern Archaeology, 19(1)*, p. 46-62.

Périn, P. (1980). *La datation des tombes mérovingiennes: historique, méthodes, applications.* Genève: Droz.

Petrie, W. M. F. (1899). Sequences in Prehistoric Remains. *The Journal of the Anthropological Institute of Great Britain and Ireland, 29, 3/4*, p. 295-301.

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/ [6.1.2023].

Ringrose, T. (2022). Package 'cobootcrs': Bootstrap Confidence Regions for Simple and Multiple Correspondence Analysis. Vers. 2.1.0 (2.3.2022). https://cran.r-project.org/web/packages/cabootcrs/cabootcrs.pdf [6.1.2023].

Sasse, B. & Theune, Cl. (1996). Perlen als Leittypen der Merowingerzeit. *Germania, 74(1)*, p. 187-231. Online: http://www.academia.edu/3710920/1996_Sasse_Theune_Perlen_als_Leittypen_der_Merowingerzeit_Germania_[6.1.2023].

Siegmund, F. (1991). Über Werkzeugspektren des Magdalénien in Mitteleuropa. *Die Kunde, NF 41/42*, p. 23-55. Online: https://www.academia.edu/13387305/%C3%9Cber_Werkzeugspektren_des_Magdal%C3%A9nien_in_Mitteleuropa [6.1.2023].

Siegmund, F. (1994). Jülich. Scherben und Schichten zu den Feuersbrünsten des 15. und 16. Jahrhunderts. *Jülicher Geschichtsblätter = Jahrbuch des Jülicher Geschichtsvereins, 62*, p. 131-184. Online: https://www.academia.edu/1285246/J%C3%BClich_Scherben_und_Schichten_zu_den_Feuersbr%C3%BCnsten_des_15._und_16._Jahrhunderts [6.1.2023].

Siegmund, F. (1995). Merovingian Beads on the Lower Rhine. *Beads = Journal of the Society of Bead Researchers, 7*, p. 37-53. Online: https://www.academia.edu/1202844/Merovingian_beads_on_the_Lower_Rhine [6.1.2023].

Siegmund, F. (1998). *Merowingerzeit am Niederrhein* (Rheinische Ausgrabungen, 34). Köln: Rheinland-Verlag.

Siegmund, F. (2013). Basel-Gasfabrik und Basel-Münsterhügel: Amphorentypologie und Chronologie der Spätlatènezeit in Basel. *Germania, 89, 2011*, p. 79-114. https://journals.ub.uni-heidelberg.de/index.php/germania/article/view/66689 [6.1.2023].

Siegmund, F. (2015). *Gewußt wie: Praxisleitfaden Seriation und Korrespondenzanalyse in der Archäologie.* Norderstedt: BoD.

Siegmund, F. (2017). Come realizzare un'analisi delle corrispondenze: guida breve per archeologi. In J. Pinar Gil (ed), *'Small finds' e Cronologia (V-IX secc.). Esempi, metodi e risultati.* (p. 31-70). Roma: BraDypUS.

Siegmund, F. (2020). *Statistik in der Archäologie: eine anwendungsorientierte Einführung auf Basis freier Software.* Norderstedt: BoD.

Smith, K. Y. & Neiman, F. D. (2007). Frequency seriation, correspondence analysis, and woodland period ceramic assemblage variation in the deep south. *Southeastern Archaeology, 26 (1)*, p. 47-72. Online: http://www.jstor.org/stable/40713417 [1.6.2023].

Stehli, P. (1973). Keramik. In Farrugia, J.-P., Kuper, R., Lüning, J. & Stehli, P. (eds). *Der bandkeramische Siedlungsplatz Langweiler 2, Gemeinde Aldenhoven, Kreis Düren.* (Rheinische Ausgrabungen 13) (p. 57-100). Bonn: Rheinland-Verlag.

**Datensätze für die praktischen Übungen:**

1a_ideal-matrix-unordered

1b_ideal-matrix-ordered

1c_ideal-matrix_longformat

2_ideal-matrix-with-one-unsensible-type

2a_ideal-matrix-with-one-unsensible-type_longformat

3_ideal-matrix-with-unspecific-grave

4_ideal-matrix-with-mixed-grave

5_ideal-matrix-with-weak-connection

6_Langweiler-2_Stehli-1973-p91-fig49

7_Schretzheim-beads_Koch-1977-table-4

8_Burt-table-from-ideal-matrix

9_Koch-U-1977-table4_xls-format

Diese Datensätze können von der Website des Autors heruntergeladen werden (www.frank-siegmund.de » Veröffentlichungen » VIII. Open Data) oder von seinem Archiv bei ResearchGate. Die Daten (Files 1… bis 8…) sind fertig vorbereitet für den Import nach PAST. Liest man die xlsx-Files mit PAST, öffnet sich unter PAST ein Menü *Import settings*, das bereits die richtigen Voreinstellungen hat, die zu diesen Datensätzen passen; daher einfach den OK-Schalter durch Anklicken bestätigen. Datensatz 9… ist als xlsx-File gespeichert und dazu gedacht, z.B. mit LO-Calc oder MS-Excel geöffnet zu werden. Die Datensätze wurden unter einer CC BY (4.0)-Lizenz veröffentlicht.

## Autor

Frank Siegmund schloss 1989 seine Dissertation über die Merowingerzeit am Niederrhein ab, in der er Korrespondenzanalysen nutzte, um eine chronologische Ordnung der Grabinventare und der Perlenketten zu erarbeiten (Siegmund 1998). Seitdem hat er diese Methode und ihre Varianten mehrfach im Kontext weiterer Studien angewandt. Wiederholt war er eingeladen, das Thema Seriation und Korrespondenzanalyse zu unterrichten und Forschungsprojekte bei der Anwendung dieser Methoden zu beraten.

### Kontakt

Priv.-Doz. Dr. Frank Siegmund

http://www.frank-siegmund.de, mail@frank-siegmund.de

### Danksagung

Der vorliegende Text ist die (sehr) stark erweiterte Fassung des Vortrags und Manuskripts *Archaeological chronologies based on correspondence analysis: a practitioner's guide to success and reliability*, präsentiert am 31. März 2014 an der Universität Bologna (Siegmund 2017). Ich danke den Organisatoren Isabella Baldini, Anna Lina Morelli und Joan Pinar Gil für ihre Einladung und allen Teilnehmern für ein sehr inspirierendes Kolloquium an einem wunderschönen Ort. Frühere Fassungen dieses Manuskripts reiften durch das Interesse, die forschende Neugierde und die kritischen Nachfragen von Studierenden in Basel, Bern, Göttingen und Münster. Mein besonderer Dank gilt Øyvind Hammer für seine freie Software PAST. Sandra Viehmeier und Christian Lau danke ich für Lektorat und Korrektorat der ersten Ausgabe 2015.