Datenanalyse

Abgabe 3

Maximilian Hagn (Matr. Nr. 11808237)

25.06.2020

1. Beispiel Multivariate Grafiken

Praxisbeispiel

Laden Sie die Daten aus der Datei ice.RData (TUWEL) in R. Die Daten beinhalten in den ersten 8 Spalten gemessene Fettsäurewerte an menschlichen Mumien. Die Spalte code unterteilt die Objekte in 6 Gruppen, und Spalte group gibt eine genauere Beschreibung der Objekte. Z.B. ist iceman unser guter alter Ötzi, glac bezieht sich auf Gletscherleichen, fresh auf "frische" Leichen (Details im Buch Varmuza und Filzmoser, 2009).

Erstellen Sie einen Segmentplot und einen Sternenplot (stars()) der ersten 8 Spalten der ice Daten. Verwenden Sie als labels der Grafiken die Namen aus der Spalte group (die Spalte kann mit dem Befehl as.character in eine Zeichenkette umgewandelt werden). Achten Sie darauf, dass Sie eine Legende hinzufügen und dass die Grafiken, die Beschriftungen und die Legende gut sichtbar sind.

Zeichnen Sie außerdem einen Plot mit parallelen Koordinaten (Funktion parcoord im Package MASS) der ersten 8 Spalten der ice Daten und färben Sie die Linien nach der Spalte code. Versuchen Sie, durch Umsortieren der Spalten die Strukturen besser sichtbar zu machen.

Daten laden

```
load(file = "ice.RData")
```

Namen laden und Farben vergeben

```
group <- data.frame(lapply(ice[10], as.character), stringsAsFactors=FALSE)
colors <- c("red","green","blue","black","brown","yellow","grey","orange")
```

Segmentplot zeichnen

```
sgmtPlot = stars(ice[1:8], labels = unlist(group), draw.segments = TRUE, key.loc = c(15,2))
```

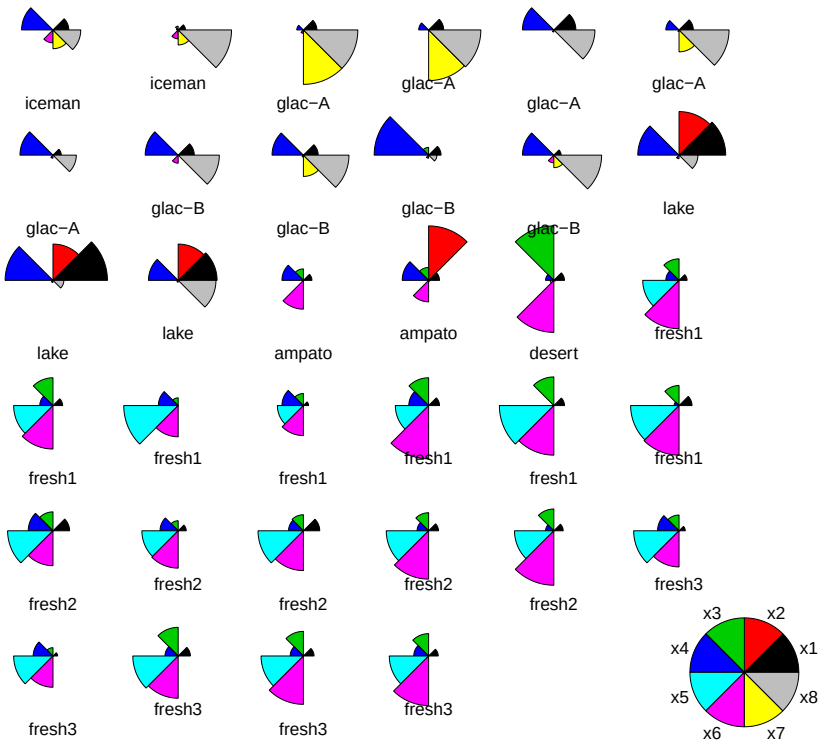


Figure 1: Segmentplot

Sternenplot zeichnen

```
starPlot = stars(ice[1:8], labels = unlist(group), lwd = 2, key.loc = c(15,2))
```

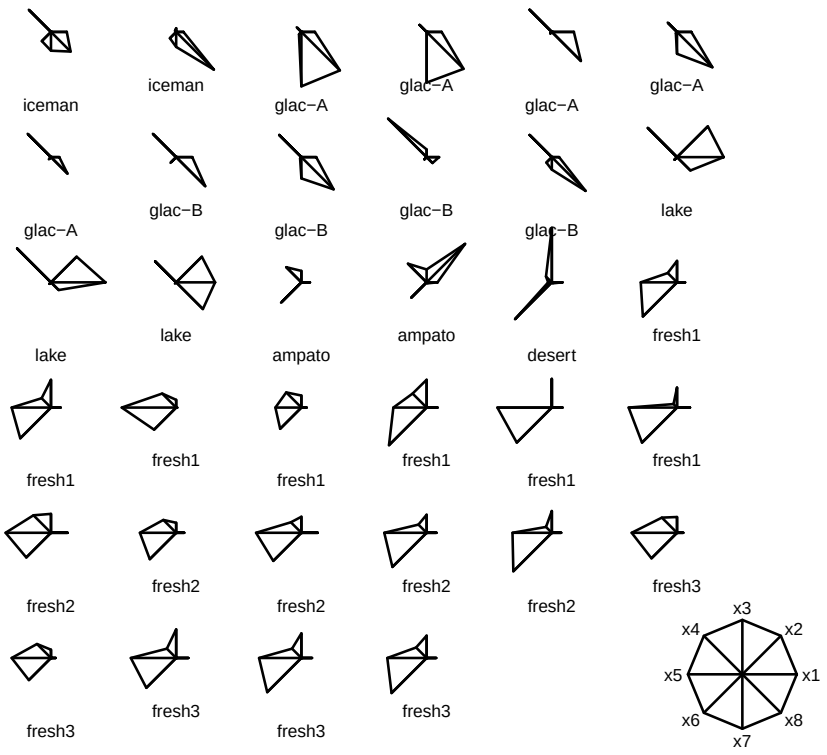


Figure 2: Sternenplot

Plot mit parallelen Korrdinaten zeichnen

```
MASS::parcoord(ice[, c(7, 2, 8, 1, 4, 3, 6, 5)], col = unlist(ice[9]))
```

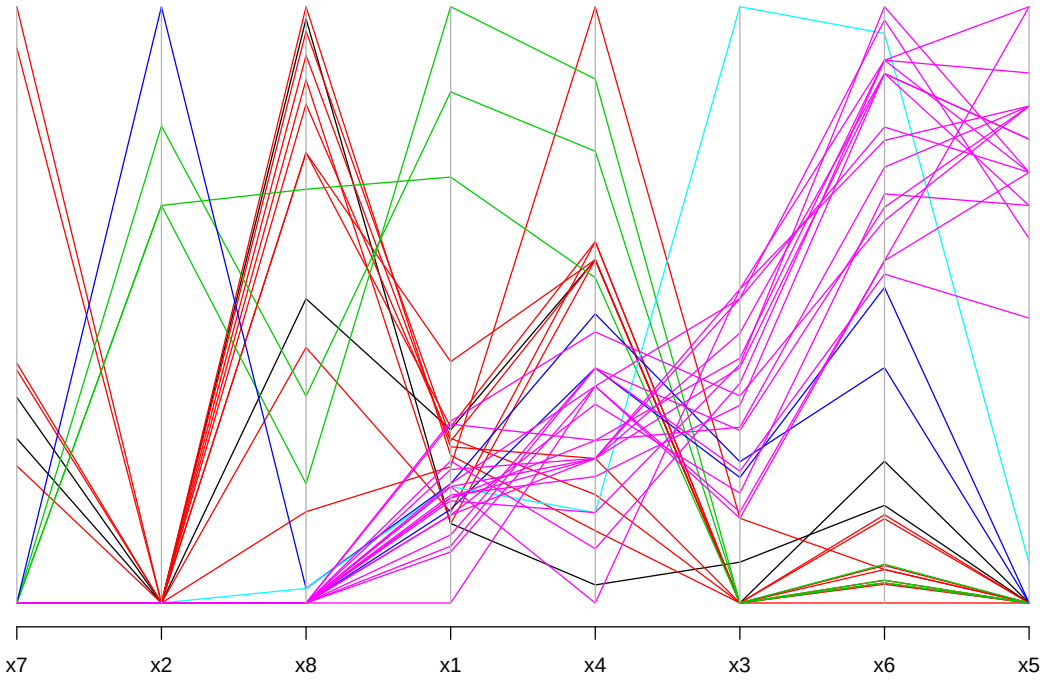


Figure 3: Plot mit parallelen Koordinaten

Fragestellungen

Welche Gemeinsamkeiten zwischen den einzelnen Mumien sehen Sie im Segment- oder Sternenplot, und korrespondieren diese Strukturen mit der Spalte group? Wird dies im Plot mit parallelen Koordinaten besser sichtbar?

Meiner Ansicht nach, ist gut zu erkennen, dass Segment- und Sternenplot die selben Daten darstellen. Vor allem bei größeren Flächen sind Übereinstimmungen zu erkennen. Bei dem Plot mit parallelen Korrdinaten werden die Gruppen mittels Farben dargestellt. Dort ist vor allem bei größeren Gruppen eine Ähnlichkeit in der Form zu erkennen.

2. Beispiel Hauptkomponmentenanalyse

Praxisbeispiel

Führen Sie eine Hauptkomponmentenanalyse (Principal Component Analysis, PCA) für den Datensatz data(decathlon) aus dem Paket FactoMineR durch. Verwenden Sie jedoch eine individuell generierte Teilmenge aus dem Datensatz. Diese wird wie folgt erstellt: decathlon[c(“Rank”, “Rank”, “Pole.vault”, “Javeline”, “Long.jump”, “Shot.put”, “Discus”, “Points”, “400m”)]

Werfen Sie einen Blick auf das caret und PCA Paket.

Daten laden

```
library("FactoMineR")
data(decathlon)
```

Hauptkomponentenanalyse erstellen

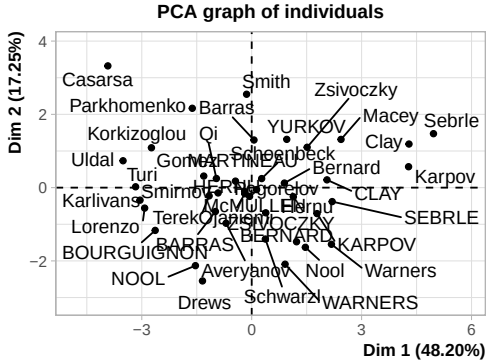
```
tm <- decathlon[c("Rank", "Rank", "Pole.vault",
                  "Javeline", "Long.jump", "Shot.put",
                  "Discus", "Points", "400m")]
```

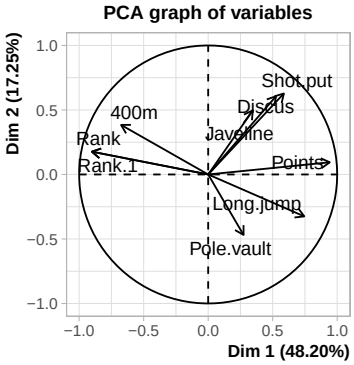
Hauptkomponentenanalyse zeichen

```
par(c(3,1))
```

```
## NULL
```

```
PCA(tm)
```






```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 41 individuals, described by 9 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

Fragestellungen

Erklären Sie den Sinn von PCA und kurz den mathematischen Hintergrund.

Bei der PCA analyse geht es darum, die Komponenten aus dem Datensatz möglichst ohne Informationsverlust zu reduzieren. Somit kann ein großer komplexer Datensatz vereinfacht dargestellt werden.

Sie basiert auf dem Mathematische Konzept der Linearkombinationen. So wird in einem großen Datensatz versucht eine Gerade zwischen den Variablenwerten zu erkennen und ein lineares Muster zu finden, dass den Datensatz bestmöglich beschreibt.

Warum sollte ggf. eine Transformation der Daten durchgeführt werden?

Die PCA kann nur verwendet werden, wenn die Daten zumindest intervallskaliert und annähernd Normalverteilt sind. Des Weiteren können so irrelevante Daten ausgeschlossen werden.

Welche Arten von Transformationen bieten sich hier an?

Daten können durch Verschiebungen an der y-Achse in eine Normalverteilung gebracht werden. Des Weiteren können Daten durch Logarithmusfunktionen verändert werden.

Wie kann eine geeignete Auswahl der Komponenten a) grafisch b) mit R (bzw. R-Funktionen) erfolgen?

3. Beispiel Clusteranalyse

Praxisbeispiel

Führen Sie für folgenden Datensatz olive aus dem Paket pgmm Clusteranalysen mit unterschiedlichen Verfahren durch. Die Daten finden Sie in ihrem Ordner oliveSample_11808237.RData. Lesen Sie sie in R ein (siehe R Hilfe des ersten Beispiels). Achten Sie darauf, die Daten passend zu visualiseren. Hierbei können Sie sich von Inspiration Visualisierung inspirieren lassen. Stellen Sie dabei das Clustering Ergebnis ähnlich wie Beispiel Clustering den realen und korrekten Gruppen gegenüber. Wenden Sie 3 unterschiedliche CLusterverfahren an und vergleichen Sie die Ergebnisse. Begründen Sie warum Datenpunkte je nach Verfahren in verschiedene Gruppen eingeordnet werden können.

Daten Laden

```
##
## ---------------------
## Welcome to dendextend version 1.13.4
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':
##
##     cutree
```

```
library("pgmm")
load(file = "oliveSample_11808237.RData")

olive <- dist(sampledData)
olive_names <- names(sampledData)
```

Cluster mit Methode Average erstellen

```
cluster <- hclust(olive, method = "average")

dendrogram <- as.dendrogram(cluster)
dendrogram <- rotate(dendrogram, 1:400)
dendrogram <- color_branches(dendrogram, k=10)

labels(dendrogram) <- paste(as.character(iris[,5])
                            [order.dendrogram(dendrogram)],"(",labels(dendrogram),")")

dendrogram <- hang.dendrogram(dendrogram,hang_height=0.1)
dendrogramaverage <- set(dendrogram, "labels_cex", 0.5)
```

Cluster mit Methode Centeroid erstellen

```
olive <- dist(sampledData)
olive_types <- names(sampledData)
cluster <- hclust(olive, method = "centroid")

dendrogram <- as.dendrogram(cluster)
dendrogram <- rotate(dendrogram, 1:400)
dendrogram <- color_branches(dendrogram, k=10)

labels(dendrogram) <- paste(as.character(iris[,5])
                            [order.dendrogram(dendrogram)],"(",labels(dendrogram),")")

dendrogram <- hang.dendrogram(dendrogram,hang_height=0.1)
dendrogramcentroid <- set(dendrogram, "labels_cex", 0.5)
```

Cluster mit Methode Median erstellen

```
olive <- dist(sampledData)
olive_types <- names(sampledData)
cluster <- hclust(olive, method = "median")

dendrogram <- as.dendrogram(cluster)
dendrogram <- rotate(dendrogram, 1:400)
dendrogram <- color_branches(dendrogram, k=10)

labels(dendrogram) <- paste(as.character(iris[,5])
                            [order.dendrogram(dendrogram)],"(",labels(dendrogram),")")

dendrogram <- hang.dendrogram(dendrogram,hang_height=0.1)
dendrogrammedian <- set(dendrogram, "labels_cex", 0.5)
```

Cluster mit Methode Average zeichen

```
plot(dendrogramaverage,
     main = "Olive Cluster Average",
     horiz =  TRUE,  nodePar = list(cex = .007))
legend("topleft", legend = olive_names)
```

Cluster mit Methode Centroid zeichen

```
plot(dendrogramcentroid,
     main = "Olive Cluster Centroid",
     horiz =  TRUE,  nodePar = list(cex = .007))
legend("topleft", legend = olive_names)
```

Cluster mit Methode Median zeichen

```
plot(dendrogrammedian,
     main = "Olive Cluster Median",
     horiz =  TRUE,  nodePar = list(cex = .007))
legend("topleft", legend = olive_names)
```

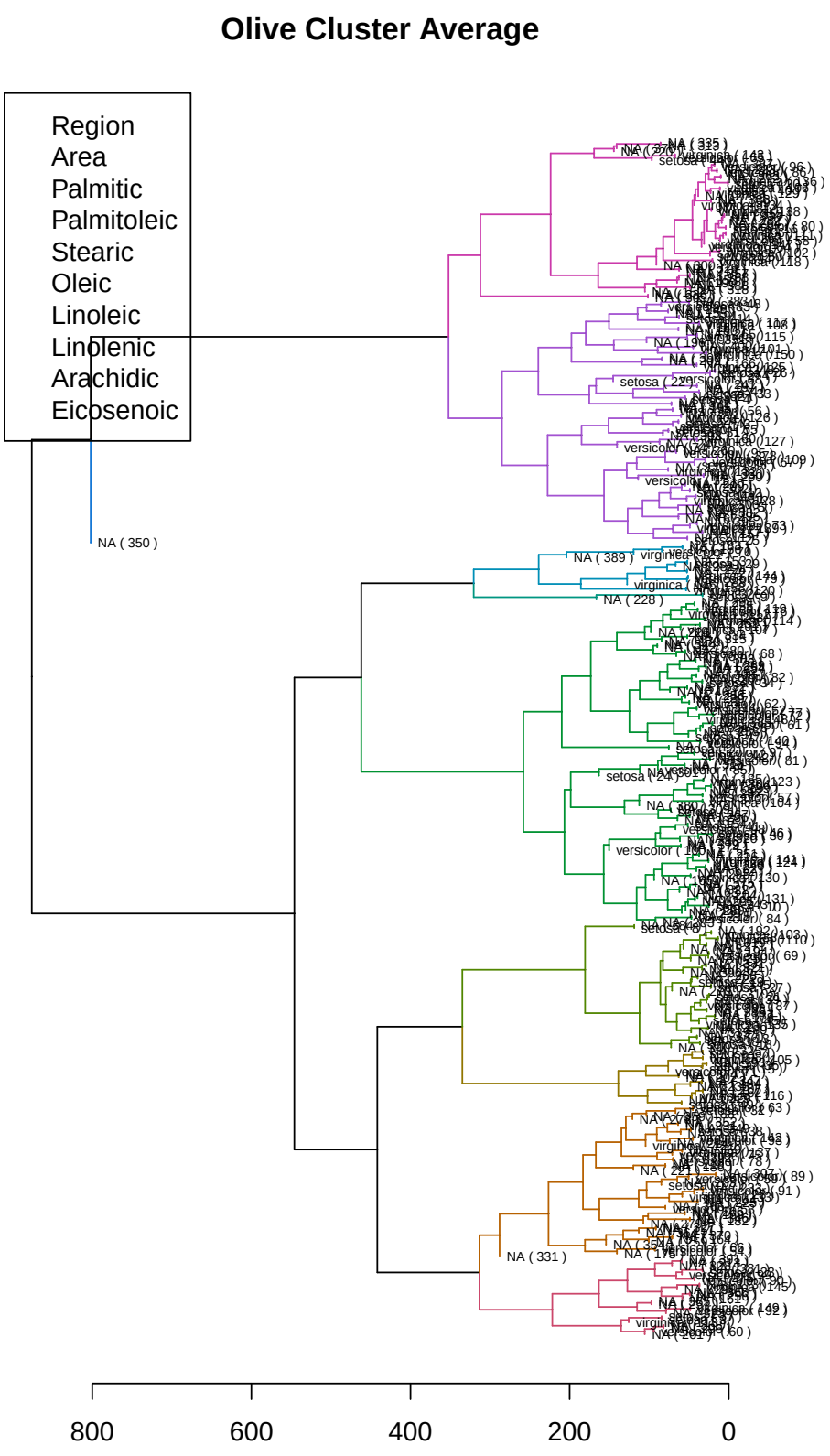


Figure 4: Olive Cluster Average

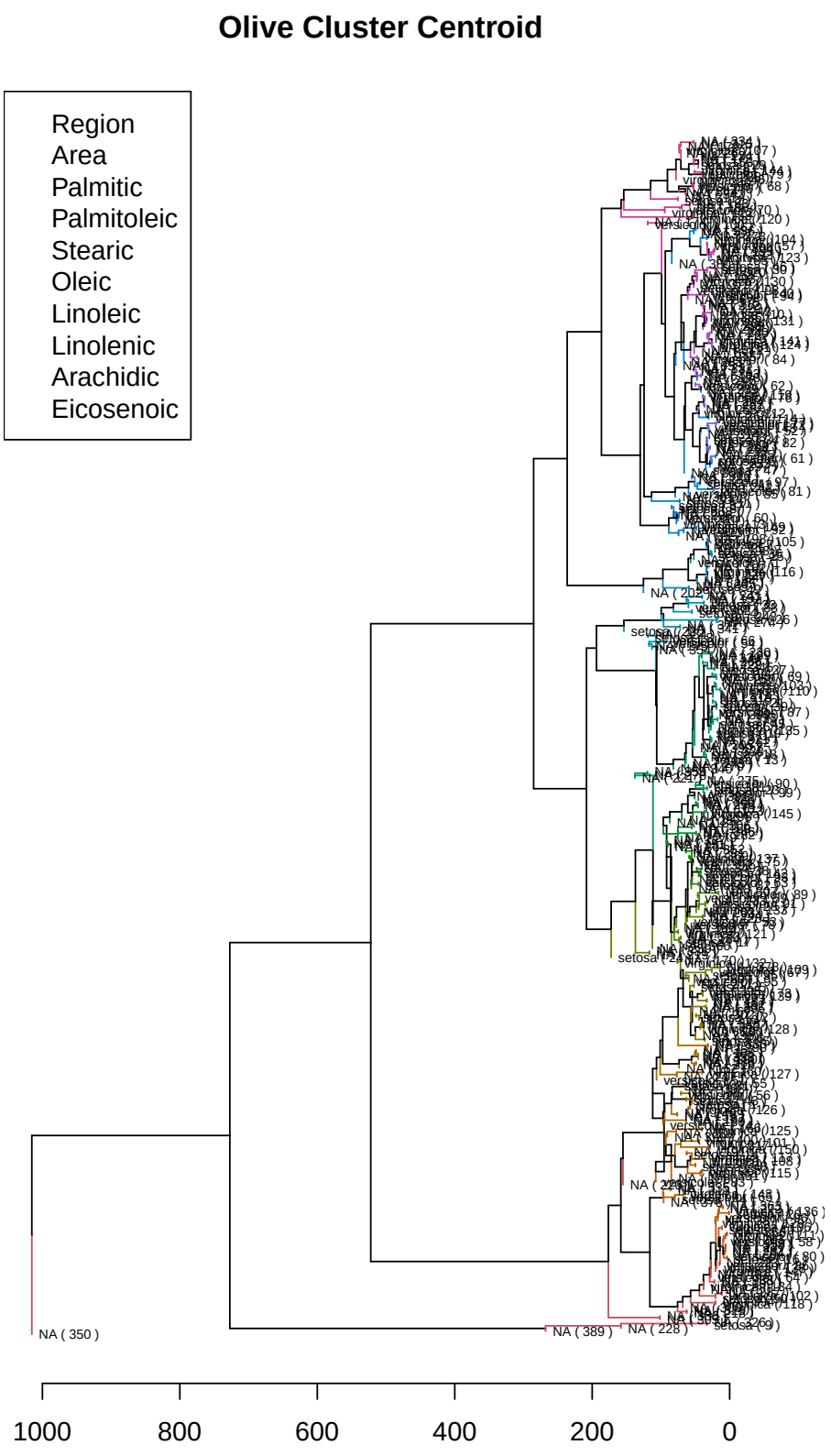


Figure 5: Olive Cluster Centroid

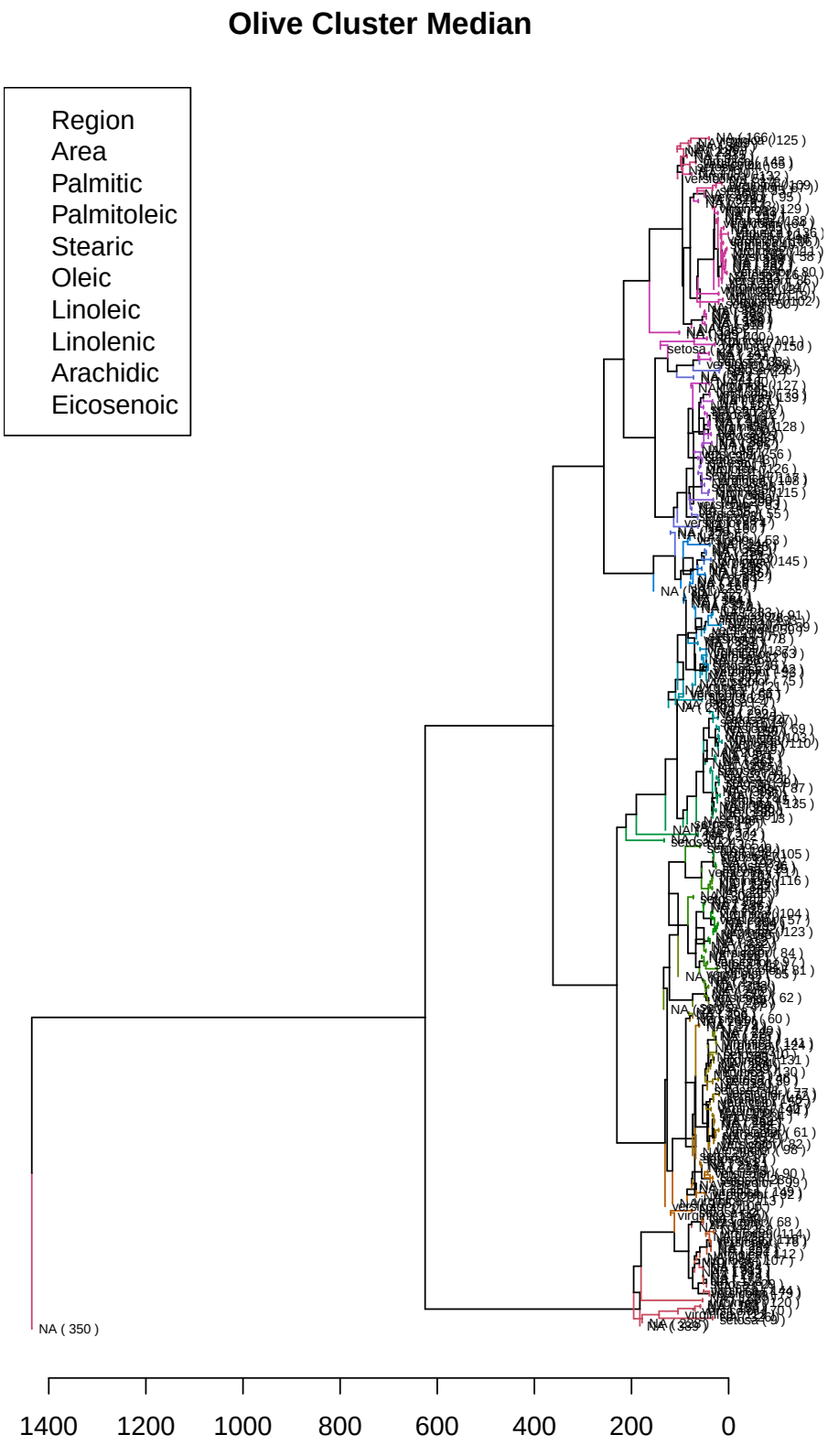


Figure 6: Olive Cluster Median

Fragestellungen

Beschreiben Sie kurz worum es bei der Clusteranalyse geht.

Bei der Clusteranalyse ist es möglich verschiedene Werte / Objekte in Gruppen einzuteilen. Ähnliche Beobachtungen sollen dabei in die gleiche Gruppe kommen und andere Beobachtungen in weitere Cluster.

Nennen Sie mindestens 5 unterschiedliche Clustering Verfahren und beschreiben Sie 2 davon kurz. Gehen Sie dabei auch auf die dazugehörigen R Funktionen ein. Erläutern Sie zusätzlich die Bedeutung von Distanzmaßen.

Partitionierungsmethoden Hierarchische Clustermethoden Complete Linkage Es wird der Maximale Abstand aller Elementenpaare beider Cluster ermittelt, dabei kann es jedoch zu kleinen Gruppen kommen Single Linkage Es wird der Minimale Abstand aller Elementenpaare beider Cluster ermittelt, dabei kann es jedoch zu Ketten kommen Average Linkage Centroid Methode Dabei wird der Abstand der Zentren beider Gruppen bestimmt. Ward Methode Fuzzy Clustering Modellbasierte Clusterung

Wie kann eine sinnnvolle und brauchbare Anzahl von Clustern bestimmt werden?

Entweder durch statistische Kriterien wie zum Beispiel ein Dendrogramm oder durch sachlogische Überlegungen.

4. Beispiel Diskriminanzanalyse

Praxisbeispiel

Diesmal werden wir folgenden Datensatz analysieren Wine. Verschaffen Sie sich zunächst einen Überblick über die Daten mit der scatterplotMatrix Funktion.

Tipp: verwenden Sie für die folgenden Ausgaben das Paket MASS und car. Welche R Funktion eignet sich, um Summary Statistics für multivariate Daten zu erstellen?

Teilen Sie den Datensatz im Verhältnis 70:30 in einen Trainings- und Testdatensatz ein. Führen Sie eine LDA für folgende Subsets des Datensatzes durch: 1) wine[c("V1", "Alcohol", "Malic", "Phenols")] 2) wine[c("V1", "Phenols", "Phenols", "Ash", "Alcalinity", "Proline", "Alcohol", "Nonflavanoid", "Malic", "Flavanoids")].

Nutzen Sie die predict Funktion zur Prognose the Klassenzugehörigkeit für die Testdaten, und Vergleichen Sie jeweils die Vorhersage mit der tatsächlichen Klasseneinteilung und visualisieren Sie das Ergebnis ensprechend.

Daten Laden

```
wine <- read.csv(file = "wine.csv")
```

–> wine.data File enthält keinen Header mit Namen, auch durch Dateinamenänderung auf .csv nicht behoben.

```
#MASS::lda(wine, c(1:177))
```

Fragestellungen

Beschreiben Sie den Nutzen sowie die Funktionsweise der Diskriminanzanalyse kurz. Grenzen Sie hierbei insbesondere Diskriminanzanalyse von Clusteranalyse ab.

Genauso wie bei der Clusteranalyse werden bei der Diskriminanzanalyse Daten gruppiert. Der Unterschied zur Clusteranalyse ist, dass nicht bekannt ist welche Beobachtungen in welche Gruppen gehören und wie viele Gruppen überhaupt existieren könnten. Als Input erhält die Analyse nur die Anzahl der Gruppen und die Klassenzugehörigkeit. Durch bestimmte Regeln soll das Verhalten vom Trainingsdatensatz auf den Datensatz rekonstruiert werden.