```
`CA` <-
   function(Y, use.svd=TRUE, color.sites="black", color.sp="red")
#
# Compute correspondence analysis (CA).
# Data table Y must contain frequencies or equivalent.
#
# use.svd: decomposition is done by svd (default). It can also be done by eigen.
#          The signs of axes may differ between the two methods.
# color  : color of the species symbols and labels in the biplots.
#
# For exercise, you may choose 'Table_9.11.txt' (small example from Chapter 9
# of the manual) or 'Spiders_28x12_spe.txt' (larger data set, real data).
#
# License: GPL-2
# Author: Pierre Legendre, January 2008
#
{
Y = as.matrix(Y)
if(min(Y) < 0) stop("Negative values not allowed in CA")
#
# Calculate three basic parameters
n = nrow(Y)
p = ncol(Y)
#
# Save the row and column names
site.names = rownames(Y)
sp.names = colnames(Y)
#
# Construct the Qbar matrix (contributions to chi-square)
# Numerical ecology (1998), equations 9.31 and 9.32
fi. = matrix(apply(Y,1,sum),n,1)
f.j = matrix(apply(Y,2,sum),1,p)
f.  = sum(fi.)
pi. = as.vector(fi./f.)
p.j = as.vector(f.j/f.)
E = (fi. %*% f.j)/f.
Qbar = (Y - E) * E^(-0.5) / sqrt(f.)
inertia = sum(Qbar^2)
#
if(use.svd) {
   # Analyse Qbar by 'svd'
   svd.res = svd(Qbar)
   k = length(which(svd.res$d > 1e-8))
   eigenvalues = svd.res$d[1:k]^2
   U = svd.res$v[,1:k]
   Uhat = svd.res$u[,1:k]
   } else {
   # Alternative analysis or Qbar by 'eigen'
   Qbar = as.matrix(Qbar)
   QprQ.eig = eigen( t(Qbar) %*% Qbar )
   k = length(which(QprQ.eig$values > 1e-16))
   eigenvalues = QprQ.eig$values[1:k]
   U = QprQ.eig$vectors[,1:k]
   Uhat = Qbar %*% U %*% diag(eigenvalues^(-0.5))
   }
#
```

```
rel.eigen = eigenvalues/inertia
rel.cum = rel.eigen[1]
for(kk in 2:k) { rel.cum = c(rel.cum, (rel.cum[kk-1] + rel.eigen[kk])) }
#
# Construct matrices V, Vhat, F, and Fhat for the ordination biplots
V = diag(p.j^(-0.5)) %*% U
Vhat = diag(pi.^(-0.5)) %*% Uhat
F = Vhat %*% diag(eigenvalues^(0.5))
Fhat = V %*% diag(eigenvalues^(0.5))
#
out <- list(total.inertia=inertia, eigenvalues=eigenvalues,
       rel.eigen=rel.eigen, rel.cum.eigen=rel.cum, U=U, Uhat=Uhat, F=F,
       Fhat=Fhat, V=V, Vhat=Vhat, site.names=site.names, sp.names=sp.names,
       color.sites=color.sites, color.sp=color.sp, call=match.call() )
class(out) <- "CA"
out
}

`print.CA` <-
    function(x, ...)
{
    cat("\nCorrespondence Analysis\n")
    cat("\nCall:\n")
    cat(deparse(x$call),'\n')
    cat("\nTotal inertia in matrix Qbar: ",x$total.inertia,'\n')
    cat("\nEigenvalues",'\n')
    cat(x$eigenvalues,'\n')
    cat("\nRelative eigenvalues",'\n')
    cat(x$rel.eigen,'\n')
    cat("\nCumulative relative eigenvalues",'\n')
    cat(x$rel.cum.eigen,'\n')
    invisible(x)
}

`biplot.CA` <-
    function(x, scaling=12, aspect=1, cex=2, ...)
# Use aspect=NA to remove the effect of parameter 'asp' in the graphs
{
if(length(x$eigenvalues) < 2) stop("There is a single eigenvalue. No plot can be
produced.")
#
# Find the limits of CA axes 1 and 2 for the plots
V.range = apply(x$V[,1:2],2,range)
Vhat.range = apply(x$Vhat[,1:2],2,range)
F.range = apply(x$F[,1:2],2,range)
Fhat.range = apply(x$Fhat[,1:2],2,range)
#
if(scaling == 12) {
# Create a drawing window for two graphs
par(mfrow=c(1,2))
#
# Biplot, scaling type = 1: plot F for sites, V for species
# The sites are at the centroids (barycentres) of the species
# This projection preserves the chi-square distance among the sites
ranF = F.range[2,] - F.range[1,]
ranV = V.range[2,] - V.range[1,]
```

```
ran.x = max(ranF[1], ranV[1])
xmin = min(V.range[1,1], F.range[1,1]) - ran.x/8
xmax = max(V.range[2,1], F.range[2,1]) + ran.x/3
ymin = min(V.range[1,2], F.range[1,2])
ymax = max(V.range[2,2], F.range[2,2])
#
plot(x$F[,1:2], asp=aspect, pch=20, cex=cex, xlim=c(xmin,xmax), ylim=c(ymin,ymax),
xlab="CA axis 1", ylab="CA axis 2", col=x$color.sites)
text(x$F[,1:2], labels=x$site.names, cex=cex, pos=4, offset=0.5, col=x$color.sites)
points(x$V[,1:2], pch=22, cex=cex, col=x$color.sp)
text(x$V[,1:2], labels=x$sp.names, cex=cex, pos=4, offset=0.5, col=x$color.sp)
title(main = c("CA biplot","scaling type 1"), family="serif")
#
# Biplot, scaling type = 2: plot Vhat for sites, Fhat for species
# The species are at the centroids (barycentres) of the sites
# This projection preserves the chi-square distance among the species
ranF = Fhat.range[2,] - Fhat.range[1,]
ranV = Vhat.range[2,] - Vhat.range[1,]
ran.x = max(ranF[1], ranV[1])
xmin = min(Vhat.range[1,1], Fhat.range[1,1]) - ran.x/8
xmax = max(Vhat.range[2,1], Fhat.range[2,1]) + ran.x/3
ymin = min(Vhat.range[1,2], Fhat.range[1,2])
ymax = max(Vhat.range[2,2], Fhat.range[2,2])
#
plot(x$Vhat[,1:2], asp=aspect, pch=20, cex=cex, xlim=c(xmin,xmax),
ylim=c(ymin,ymax), xlab="CA axis 1", ylab="CA axis 2", col=x$color.sites)
text(x$Vhat[,1:2], labels=x$site.names, cex=cex, pos=4, offset=0.5,
col=x$color.sites)
points(x$Fhat[,1:2], pch=22, cex=cex, col=x$color.sp)
text(x$Fhat[,1:2], labels=x$sp.names, cex=cex, pos=4, offset=0.5, col=x$color.sp)
title(main = c("CA biplot","scaling type 2"), family="serif")

} else {

# Biplot, scaling type = 3: plot F for sites, Fhat for species
# This projection preserves the chi-square distance among the sites and species
ranF    = F.range[2,] - F.range[1,]
ranFhat = Fhat.range[2,] - Fhat.range[1,]
ran.x = max(ranF[1], ranFhat[1])
xmin = min(Fhat.range[1,1], F.range[1,1]) - ran.x/8
xmax = max(Fhat.range[2,1], F.range[2,1]) + ran.x/3
ymin = min(Fhat.range[1,2], F.range[1,2])
ymax = max(Fhat.range[2,2], F.range[2,2])
#
plot(x$F[,1:2], asp=aspect, pch=20, cex=cex, xlim=c(xmin,xmax), ylim=c(ymin,ymax),
xlab="CA axis 1", ylab="CA axis 2", col=x$color.sites)
text(x$F[,1:2], labels=x$site.names, cex=cex, pos=4, offset=0.5, col=x$color.sites)
points(x$Fhat[,1:2], pch=22, cex=cex, col=x$color.sp)
text(x$Fhat[,1:2], labels=x$sp.names, cex=cex, pos=4, offset=0.5, col=x$color.sp)
title(main = c("CA biplot","scaling type 3"), family="serif")
}
#
invisible()
}
```

From:
https://anadat-r.davidzeleny.net/ - **Analysis of community ecology data in R**

Permanent link:
https://anadat-r.davidzeleny.net/doku.php/cs:numecolr:ca

Last update: **2017/10/11 20:36**