Introduction

1. Principal Component Analysis (PCA), which is used to summarize the information contained in a continuous (i.e, quantitative) multivariate data by reducing the dimensionality of the data without loosing important information.
2. Correspondence Analysis (CA), which is an extension of the principal component analysis suited to analyse a large contingency table formed by two *qualitative variables* (or categorical data).
3. Multiple Correspondence Analysis (MCA), which is an adaptation of CA to a data table containing more than two categorical variables.
4. Multiple Factor Analysis (MFA) dedicated to datasets where variables are organized into groups (qualitative and/or quantitative variables).
5. Hierarchical Multiple Factor Analysis (HMFA): An extension of MFA in a situation where the data are organized into a hierarchical structure.
6. Factor Analysis of Mixed Data (FAMD), a particular case of the MFA, dedicated to analyze a data set containing both quantitative and qualitative variables.

Main functions

- Visulizing dimension reduction analysis outputs

fviz_eig: extract and visualize the eigenvalues/variances of dimensions

fviz_pca: graph of individuals/variables from the output of PCA

fviz_ca: graph of column/row variables from the output of CA

fviz_mca: graph of individuals/variables from the output of MCA

fviz_mfa: graph of individuals/variables from the output of MFA

fviz_famd: graph of individuals/variables from the output of FAMD

fviz_hmfa: graph of individuals/vairables from the output of HMFA

fviz_ellipses: draw condidence ellipses around the categories

fviz_cos2: visualize the quality of representation of the row/column variable from the results to PCA, CA, MCA functions

fviz_contrib: visualize the contribution of row/column elements from the results of PCA, CA, MCA functions

- Extracting data from dimension reduction analysis outputs

get_eigenvalue: extract and visualize the eigenvalues/variance of dimensions

get_pca: extract all the results(coordinates, squared cosine, contributions) for the active individuals/variables from PCA outputs

get_ca: extract all the results(coordinates, squared cosine, contributions) for the active column/row variables from CA outputs

get_mca: extract results from MCA outputs

get_mfa: extract results from MFA outputs

get_famd: extract results from FAMD outputs

get_hmfa: extract results from HMFA outputs

facto_summarize: subset and summarize the output of factor analyses

- Clustering analysis and visualization

dist(fviz_dist, get_dist): enhanced distance matrix computation and visualization

get_clust_tendency: assessing clustering tendency

fviz_nbclust(fviz_gap_stat): determining and visualizing the optimal number of clusters

fviz_dend: enhanced visulization of dendrogram

fviz_cluster: visualize clustering results

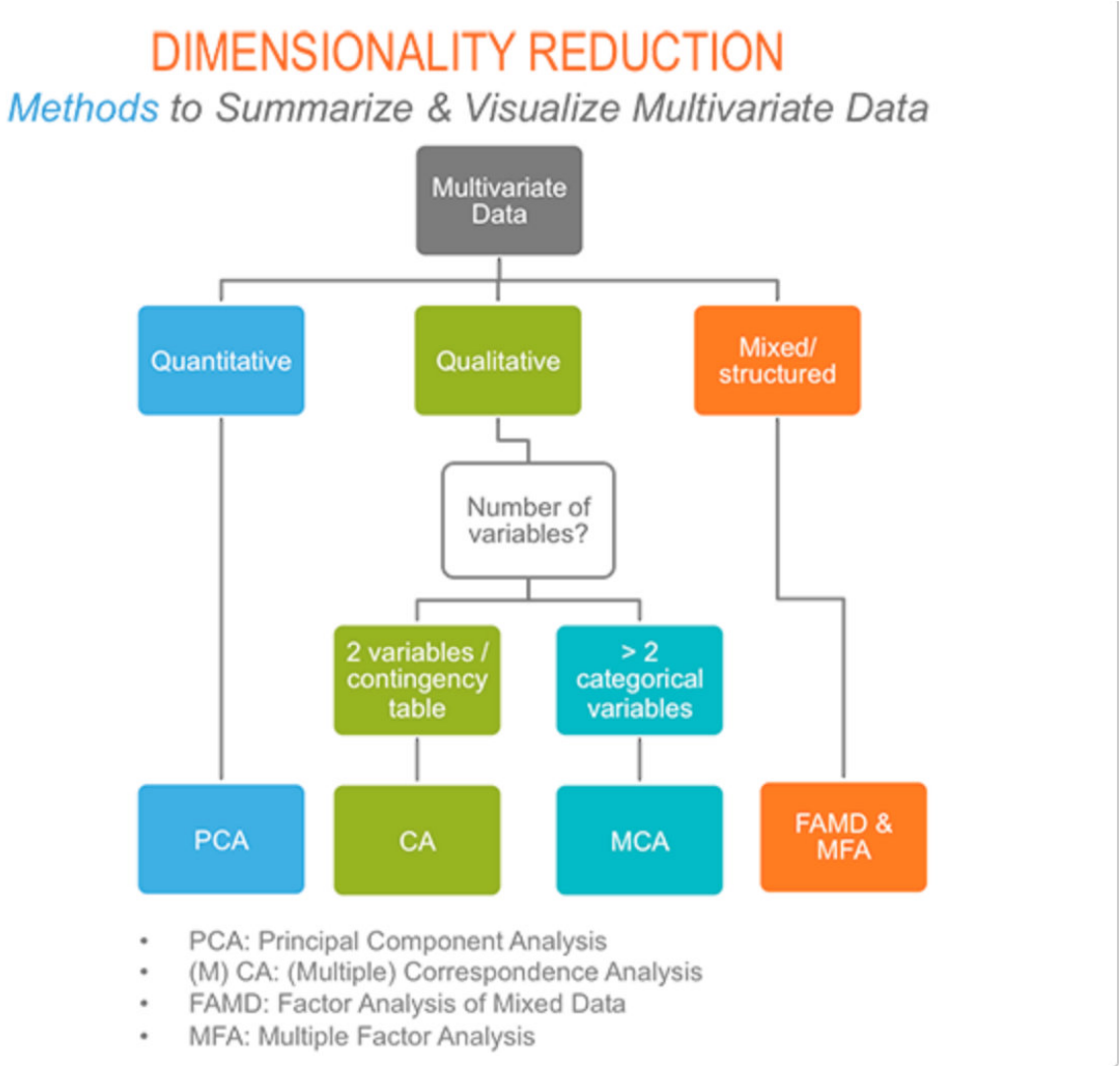
fviz_mclust: visualzie model-based clustering results

fviz_silhouette: visualize silhouette information from clustering

hcut: computes hierarchical clustering and cut the tree

hkmeans(hkmeans_tree, print.hkmeasn): hierarchical k-means clustering

eclust: visual enhancement of clustering analysis



Principal component analysis

```
library(factoextra)
```

```
data(decathlon2)
```

```
df <- decathlon2[1:23, 1:10]
```

```
library(FactoMineR)
```

```
res.pca <- PCA(df, graph=F)
```

来自FactoMinR包的PCA函数返回对应PCA各项值

```
PCA(X, scale.unit=T, ncp=5, graph=T)
```

scale.unit：根据单位变量对数据标准化，默认为T

ncp：返回对应数目的维度，默认为5

graph： 展示图形，默认为T

主要输出值

eig，包含所有eigenvalues的矩阵，variance的百分比和累积variance百分比

var，包含所有variables的矩阵，对应的coordinates，correlations，axes，square cosin，contributions

ind，包含所有individuals的矩阵，对应coordinates，square cosine，contributions

---

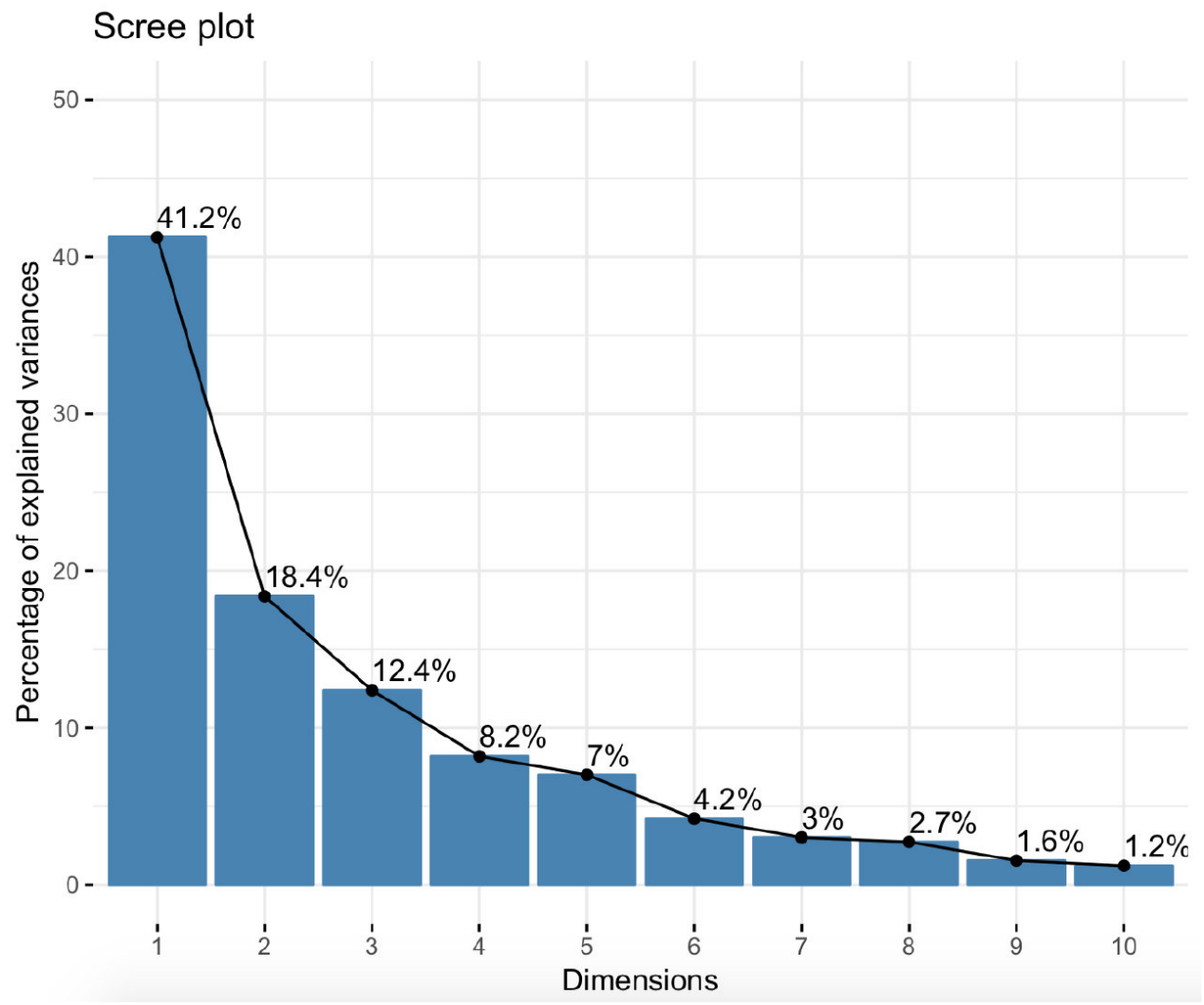
获得各主成的variables的eigenvalues

```
get_eig(res.pca)
```

```
> get_eig(res.pca)
       eigenvalue variance.percent cumulative.variance.percent
Dim.1   4.1242133        41.242133                    41.24213
Dim.2   1.8385309        18.385309                    59.62744
Dim.3   1.2391403        12.391403                    72.01885
Dim.4   0.8194402         8.194402                    80.21325
Dim.5   0.7015528         7.015528                    87.22878
Dim.6   0.4228828         4.228828                    91.45760
Dim.7   0.3025817         3.025817                    94.48342
Dim.8   0.2744700         2.744700                    97.22812
Dim.9   0.1552169         1.552169                    98.78029
Dim.10  0.1219710         1.219710                   100.00000
```

绘制各个维度对应variables的eigenvalues的碎石图

```
fviz_screeplot(res.pca, addlables=T, ylim=c(0,50))
```



提取variables的PCA结果

```
var <- get_pca_var(res.pca)
```

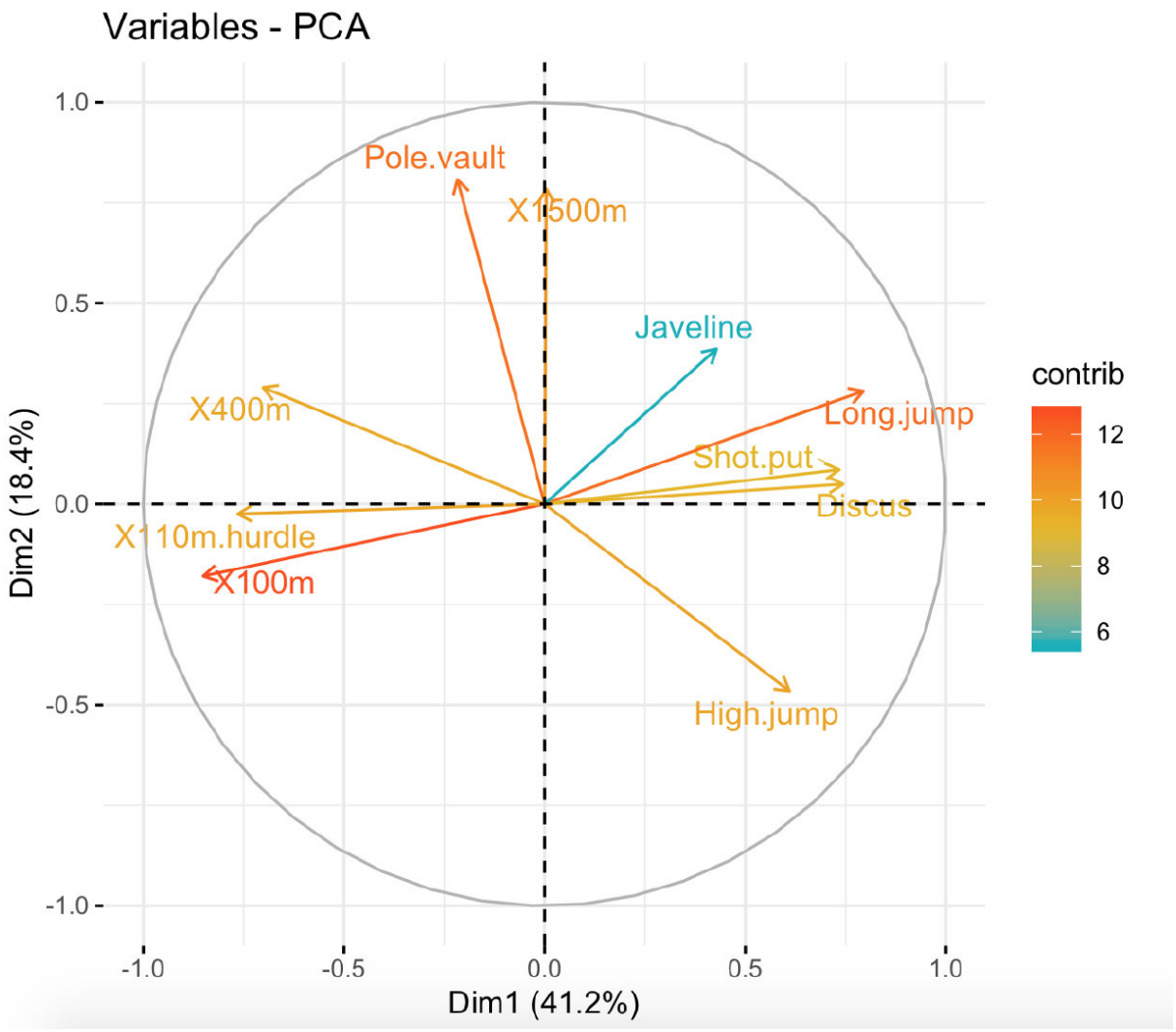
```
> var
Principal Component Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the variables"
2 "$cor"     "Correlations between variables and dimensions"
3 "$cos2"    "Cos2 for the variables"
4 "$contrib" "contributions of the variables"
```

默认绘制PCA的variables图

```
fviz_pca_var(res.pca, col.car="black")
```

或者根据对应的variables的值绘制颜色梯度图，例如根据contrib来绘制颜色梯度，对应repel=T参数避免标签文字重叠

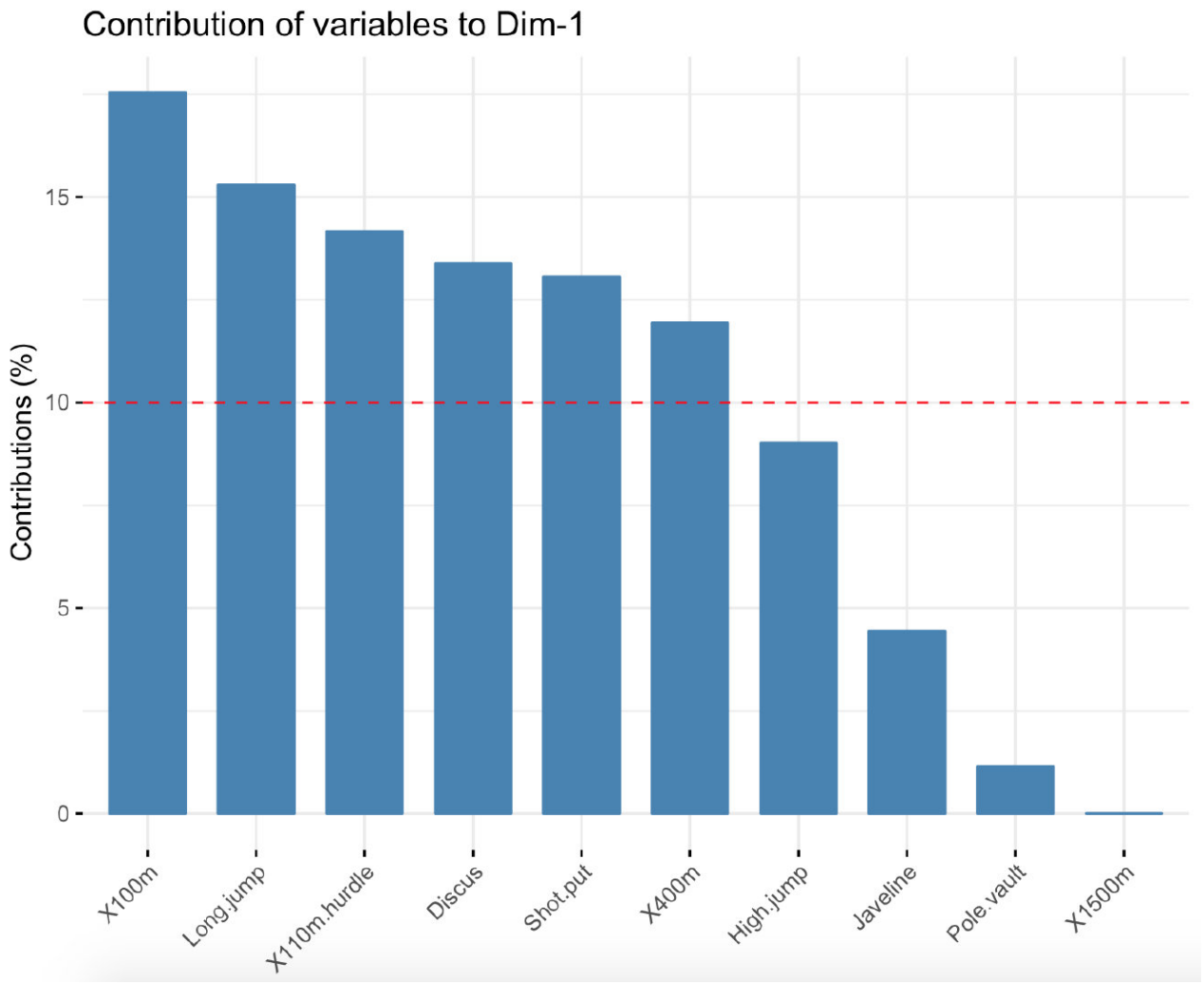
```
fviz_pca_var(res.pca, col.var="contrib", gradient.cols=c("#00AFBB", "#E7B800", "#FC4E07"), repel=T)
```



各个varaibles对各主成维度(axes)的贡献情况

维度1，PC1各variables贡献分布, choice展示PCA的var或ind，axes指定第几个维度，top指定展示variable数目

```
fviz_contrib(res.pca, choice="var", axes=1, top=5)
```



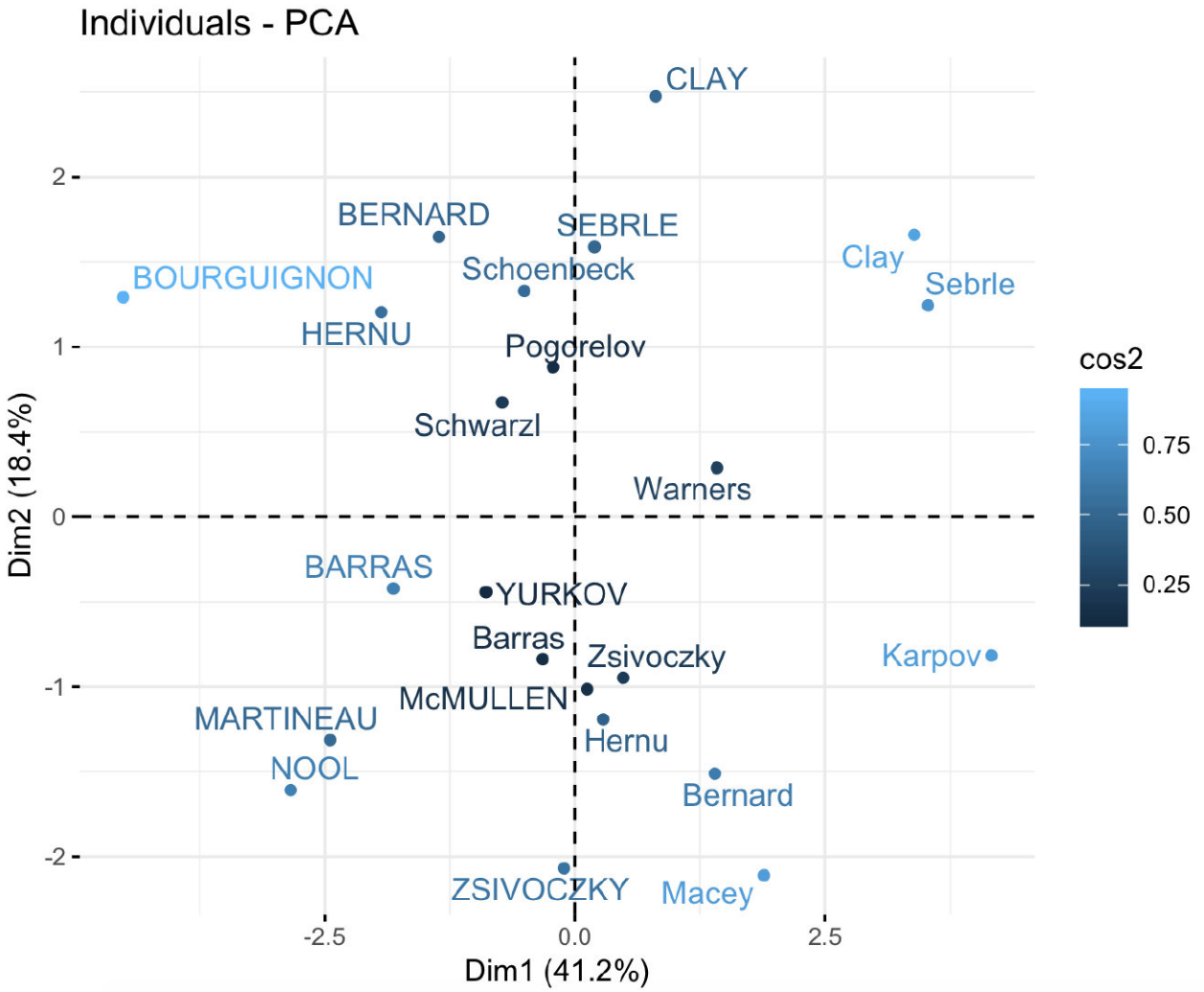
获得PCA各个individuals的结果

```
ind <- get_pca_ind(res.pca)
```

```
> ind
Principal Component Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

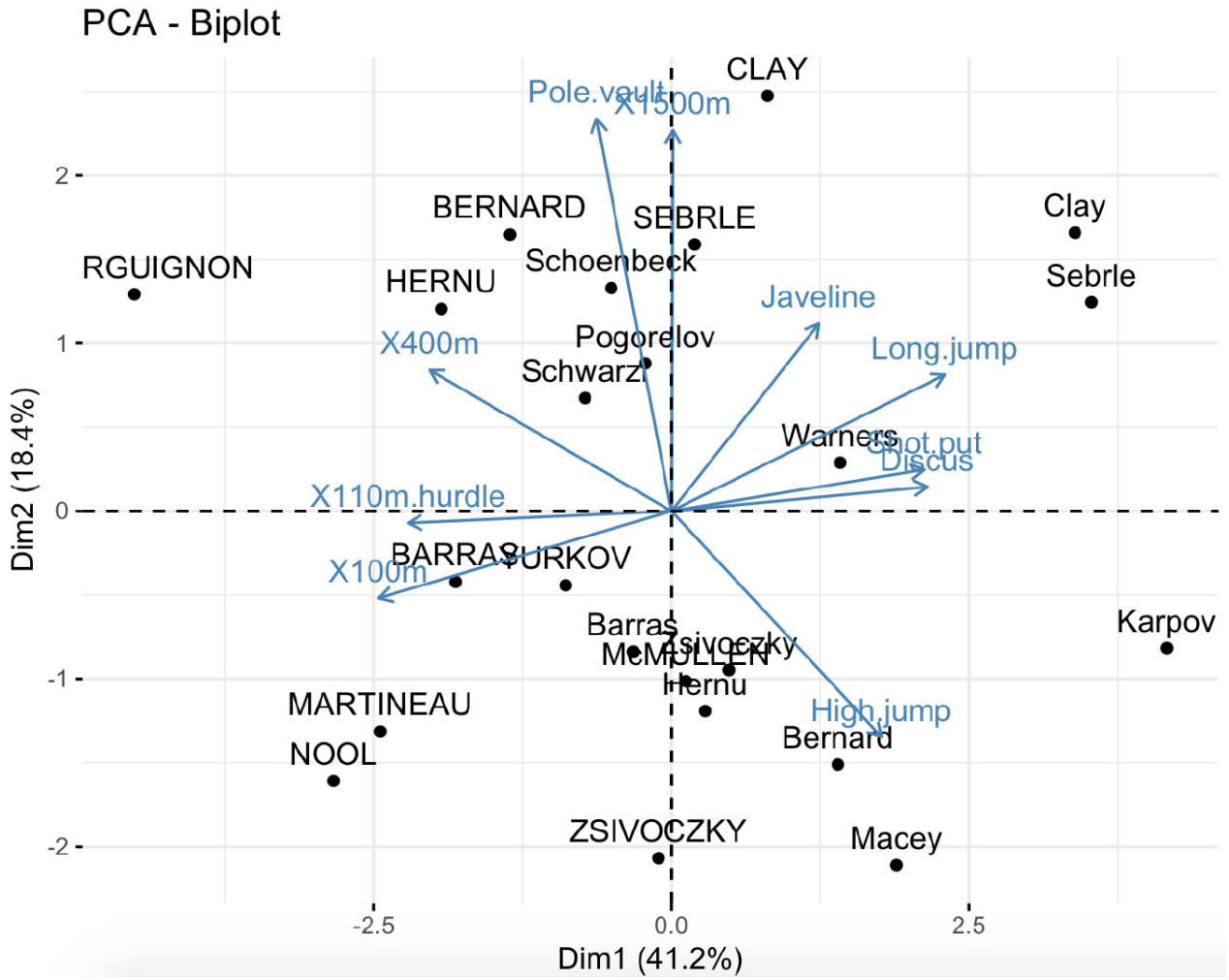
展示各个individuals主成分布情况，repel=T避免各个标签文字重叠；col.ind="cos2"，cos2对应各个individuals的cosine平方值，根据其分布绘制颜色梯度

```
fviz_pca_ind(res.pca, col.ind="cos2", gredient.cols=c("#00AFBB", "#E7B800", "#FC4E07"), reple=T)
```



使用biplot展示individuals和variables主成图

```
fviz_pca_biplot(res.pca, reple=T)
```



针对分组individuals绘制PCA图

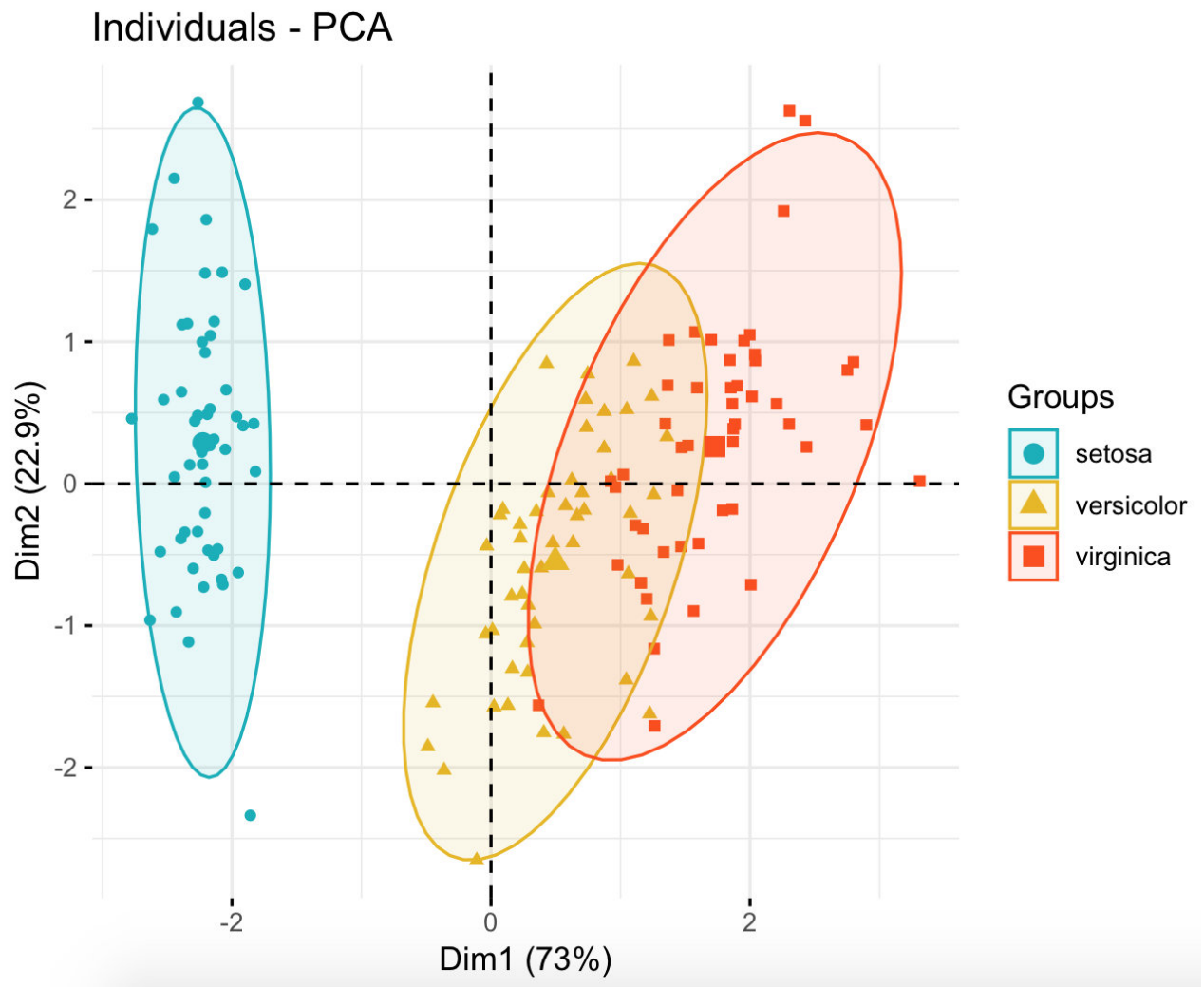
```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
iris.pca <- PCA(iris[,-5], graph=F)
```

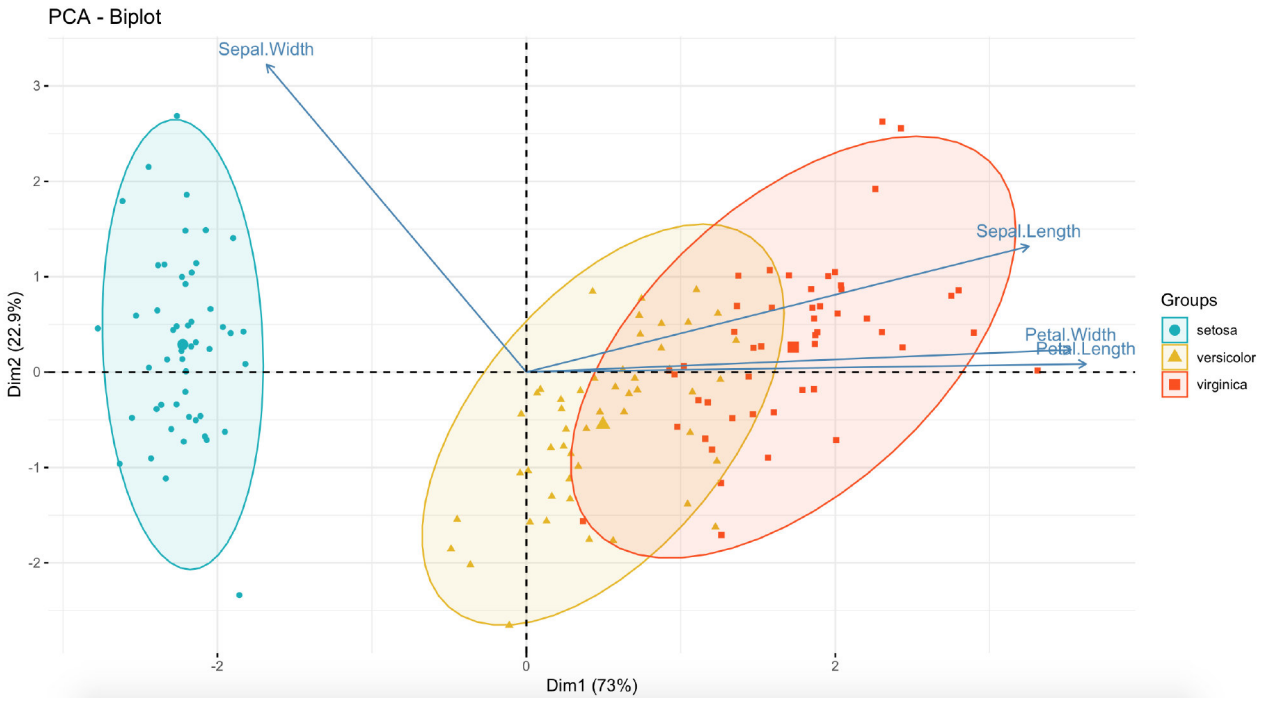
绘制PCA图，根据group信息对individuals着色区分

habillage，根据分组对观测值进行着色；palette，设定调色板范围；allEllipses=T，当habillage不为none时，围绕individuals边缘绘制椭圆

```
fviz_pca_ind(iris.pca, label="none", habillage=iris$Species, palette=c("#00AFBB", "#E7B800", "#FC4E07"), addEllipses=T)
```




```
fviz_pca_biplot(iris.pca, label="var",reple=T, habillage=iris$Species,
palette=c("#00AFBB", "#E7B800", "#FC4E07"),addEllipses=T)
```



Correspondance analysis

CA是PCA的延伸，用于研究定型变量之间的关系(或者分类变量)，同PCA，提供了在二维图形中展示和绘制数据的方法。例如，分析两分类数据的频率，也就是列联表。针对列联表的行和列提供factor scores(coordinates)，用于图示列联表行和列单元之间的相关性。

当分析two-way 列联表时，典型需要关注的问题为，是否存在一些明确的行单元和一些列单元相关。CA通过几何图形过程在低纬度的空间使用点来描绘two-way列联表的行和列，这样行和列的点的位置对应了表格中他们的相关性。

```
library(FactoMineR)
```

```
data(housetasks)
```

```
> head(housetasks)
           Wife Alternating Husband Jointly
Laundry     156           14       2       4
Main_meal   124           20       5       4
Dinner       77           11       7      13
Breakfeast   82           36      15       7
Tidying      53           11       1      57
Dishes       32           24       4      53
```

housetasks数据，行对应了不同的家庭任务，列对应家庭任务的执行成员，期间的值表示执行数量。

可以使用Chi-square test of independent分析两分类变量的频数表(列联表)，用于评估两分类变量关系的显著性。

使用gplots包中的balloonplot函数图形化展示列联表

转换成表格

```
dt <- as.table(as.matrix(housetasks))
```

图形展示

```
balloonplot(t(dt), main="housetasks", xlab="", ylab="", label=F, show.margins=F)
```

针对小的列联表，可以直接使用Chi-square test评估行和列分类之间的显著相关性

```
chisq <- chisq.test(housetasks)
```

可见行和列变量具有显著相关性(p-value=r chisq$p-value)

使用FactoMinR函数CA对行和列的点分析

默认，ncp=5，保留5个dimensions；graph，是否展示图；row.sup和col.sup对应数组向量指定行和列的矩阵

```
res.ca <- CA(housetasks, graph=F)
```

```
> res.ca
**Results of the Correspondence Analysis (CA)**
The row variable has  13  categories; the column variable has 4 categories
The chi square of independence between the two variables is equal to 1944.456 (p-value =  0 ).
*The results are available in the following objects:

   name              description
1  "$eig"            "eigenvalues"
2  "$col"            "results for the columns"
3  "$col$coord"      "coord. for the columns"
4  "$col$cos2"       "cos2 for the columns"
5  "$col$contrib"    "contributions of the columns"
6  "$row"            "results for the rows"
7  "$row$coord"      "coord. for the rows"
8  "$row$cos2"       "cos2 for the rows"
9  "$row$contrib"    "contributions of the rows"
10 "$call"           "summary called parameters"
11 "$call$marge.col" "weights of the columns"
12 "$call$marge.row" "weights of the rows"
```

和PCA分析类似，可以通过get_ca_row/col获得行和列变量结果

```
row <- get_ca_row(res.ca)
```

```
> row
Correspondence Analysis - Results for rows
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the rows"
2 "$cos2"    "Cos2 for the rows"
3 "$contrib" "contributions of the rows"
4 "$inertia" "Inertia of the rows"
> col <- get_ca_col(res.ca)
> col
Correspondence Analysis - Results for columns
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the columns"
2 "$cos2"    "Cos2 for the columns"
3 "$contrib" "contributions of the columns"
4 "$inertia" "Inertia of the columns"
```

row和col所包含数据可用于绘制对应图形

row$coord，每个行点在每个维度的坐标，用于绘制scatter plot

row$cos2，行所代表的性能，也就是quality on the factor map，每个dimensions中行的所有的cos2之和为1

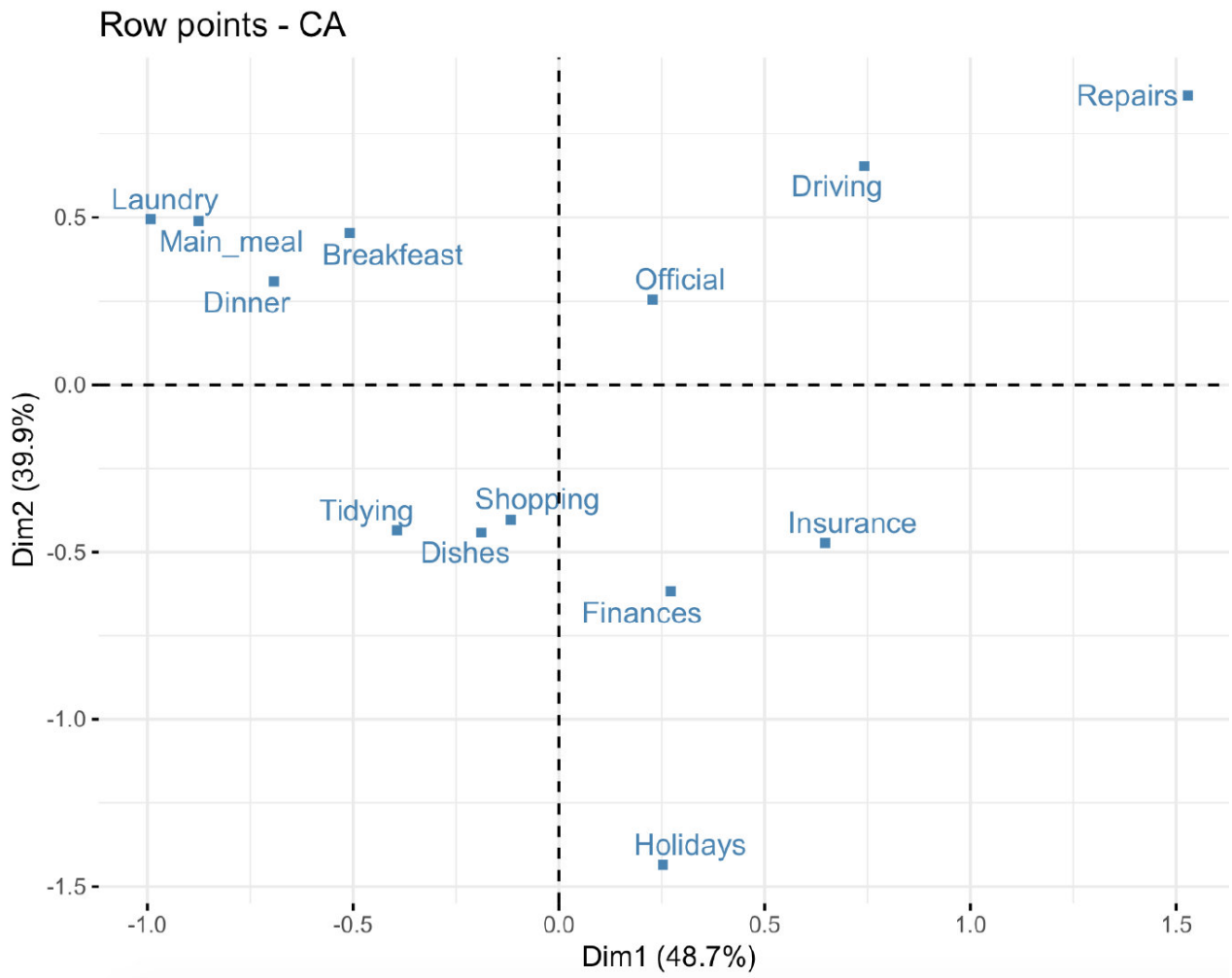
the quality of representation of the rows on the factor map is called the squared cosine(cos2) or the squared correlations

row$contrib，行对应每个维度的贡献值

绘制行点图

col.row定义颜色，shape.row定义形状

```
fviz_ca_row(res.ca, col.row="steelblue", shape.row=15, repel=T)
```



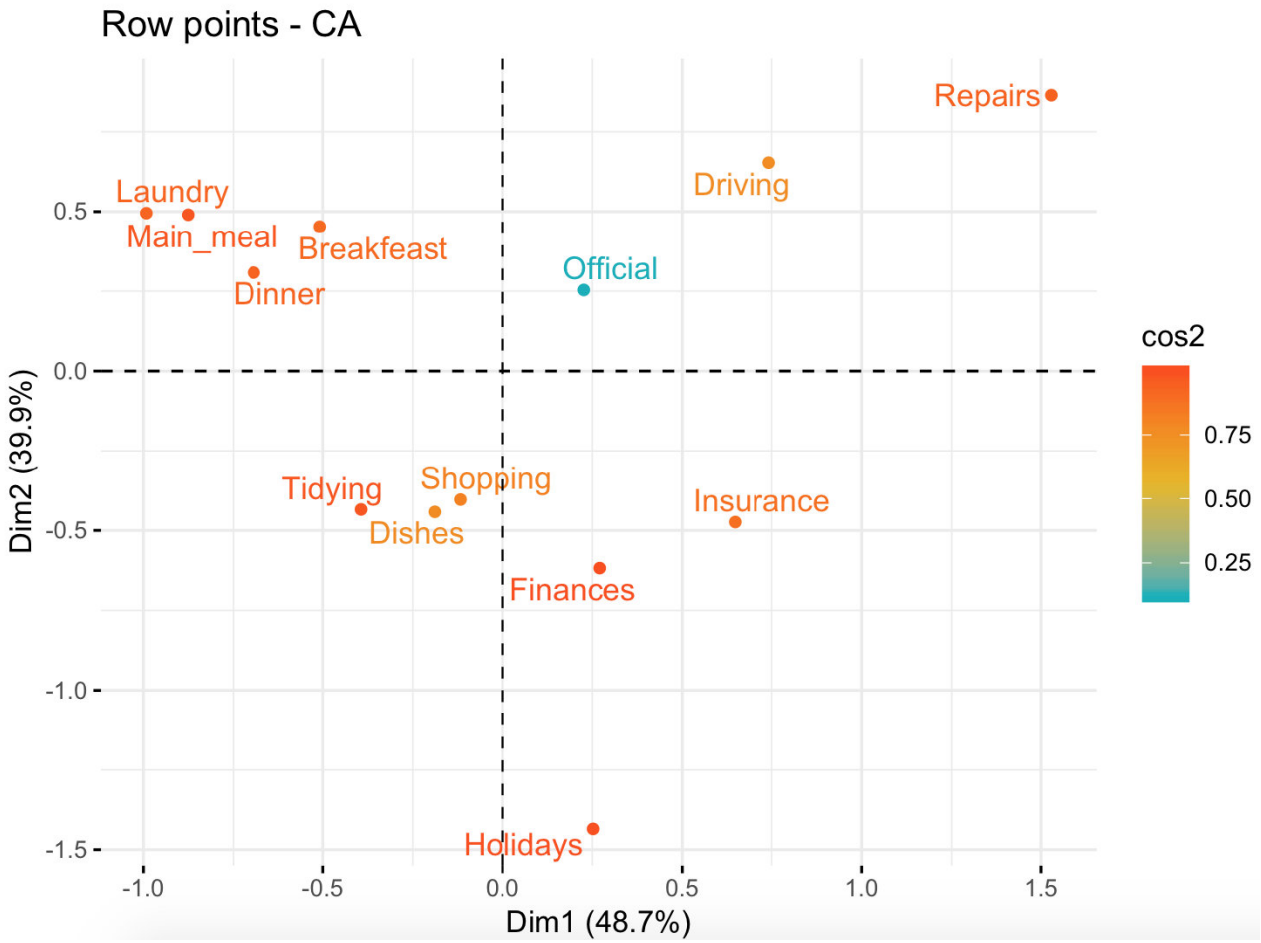
该图描绘了行单元点之间的关系，具有类似特点的行单元点聚集一起；负相关的行单元点位于图像相对的象限；不同点之间的距离和原点对应行quality of the row points on the factor map，远离原点的行单元点更能很好代表factor map

根据cos2对行单元点绘图：**cos2代表了行/列和对应axis(dimensions)之间的相关程度**

```
> head(row$cos2,4)
                Dim 1     Dim 2      Dim 3
Laundry    0.7399874 0.1845521 0.07546047
Main_meal  0.7416028 0.2323593 0.02603787
Dinner     0.7766401 0.1537032 0.06965666
Breakfeast 0.5049433 0.4002300 0.09482670
```

这里低cos2着白色，中等cos2着蓝色，高cos2着红色

```
fviz_ca_row(res.ca, col.row="cos2", gradient.cols=c("#00AFBB", "#E7B800", "#FC4E07"), repel=T)
```



绘制列单元点图

```
fviz_ca_col(res.ca, col.col="cos2", gradient.cols=c("#00AFBB", "#E7B800", "#FC4E07"), repel=T)
```

使用corrplot包绘制各维度cos2贡献值图

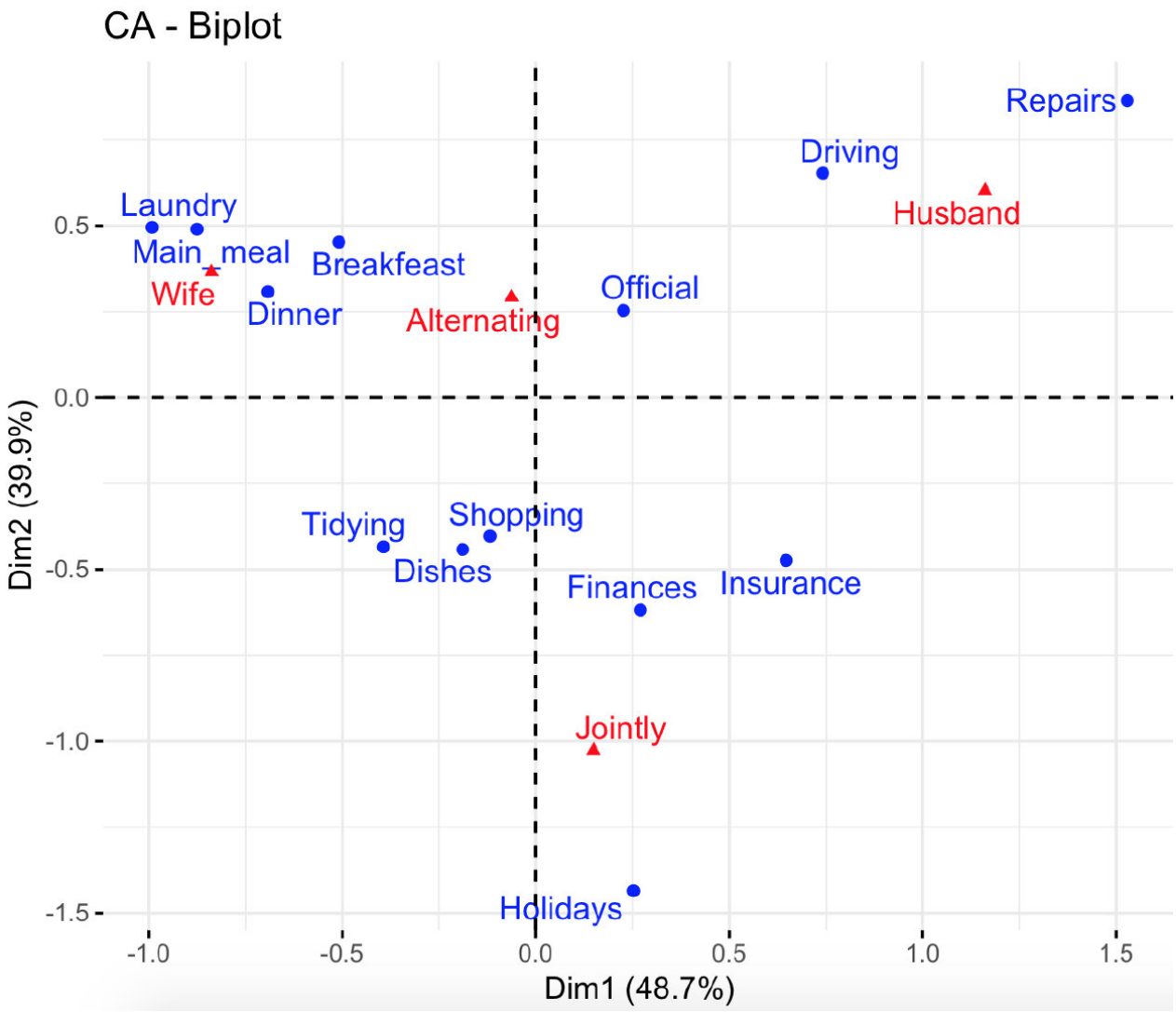
```
library(corrplot)
```

```
corrplot(row$cos2, is.corr=F)
```

略

简单绘制biplot图

```
fviz_ca_biplot(res.ca, repel=T)
```



Mutiple correspondence analysis

MCA是CA的扩展，展示超过两个分类变量之间的关系。可以看作为针对类别而不是数量做PCA。

MCA用于识别，同一个解答中具有类似特征的一组变量；分类变量之间的相关性。

数据来自关于小学儿童食物中毒的调查，查询所吃食物和对应的病症

```
data(poison)
```

数据第1/2列对应年龄和时间；列3/4对应是否疾病和性别，用于对分组个体着色

```
res.mca <- MCA(poison.active, graph=F)
```

```
> res.mca
**Results of the Multiple Correspondence Analysis (MCA)**
The analysis was performed on 55 individuals, described by 11 variables
*The results are available in the following objects:

   name              description
1  "$eig"            "eigenvalues"
2  "$var"            "results for the variables"
3  "$var$coord"      "coord. of the categories"
4  "$var$cos2"       "cos2 for the categories"
5  "$var$contrib"    "contributions of the categories"
6  "$var$v.test"     "v-test for the categories"
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"           "intermediate results"
12 "$call$marge.col" "weights of columns"
13 "$call$marge.li"  "weights of rows"
```

通过factoextra函数获得对应值

get_eigenvalue(res.mca)，获得每个dimension的eigenvalues/variances

fviz_eig(res.mca)，绘制eigenvalues/variances图

get_mca_ind(res.mca), get_mca_var(res.mca)，获得individuals和variables值

fviz_mca_ind(res.mca), fviz_mca_var(res.mca)，绘制individuals和variables图

fviz_mca_biplot(res.mca)，绘制行和列的biplot

获得eigenvalues值绘制碎石图

```
eig.val <- get_eig(res.mca)
```

```
fviz_screeplot(res.mca, addlabels=T,ylim=c(0,45))
```

获取MCA的individuals/variables值

```
var <- get_mca_var(res.mca)
```

```
ind <- get_mca_ind(res.mca)
```

```
> var
Multiple Correspondence Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for categories"
2 "$cos2"    "Cos2 for categories"
3 "$contrib" "contributions of categories"
> ind
Multiple Correspondence Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

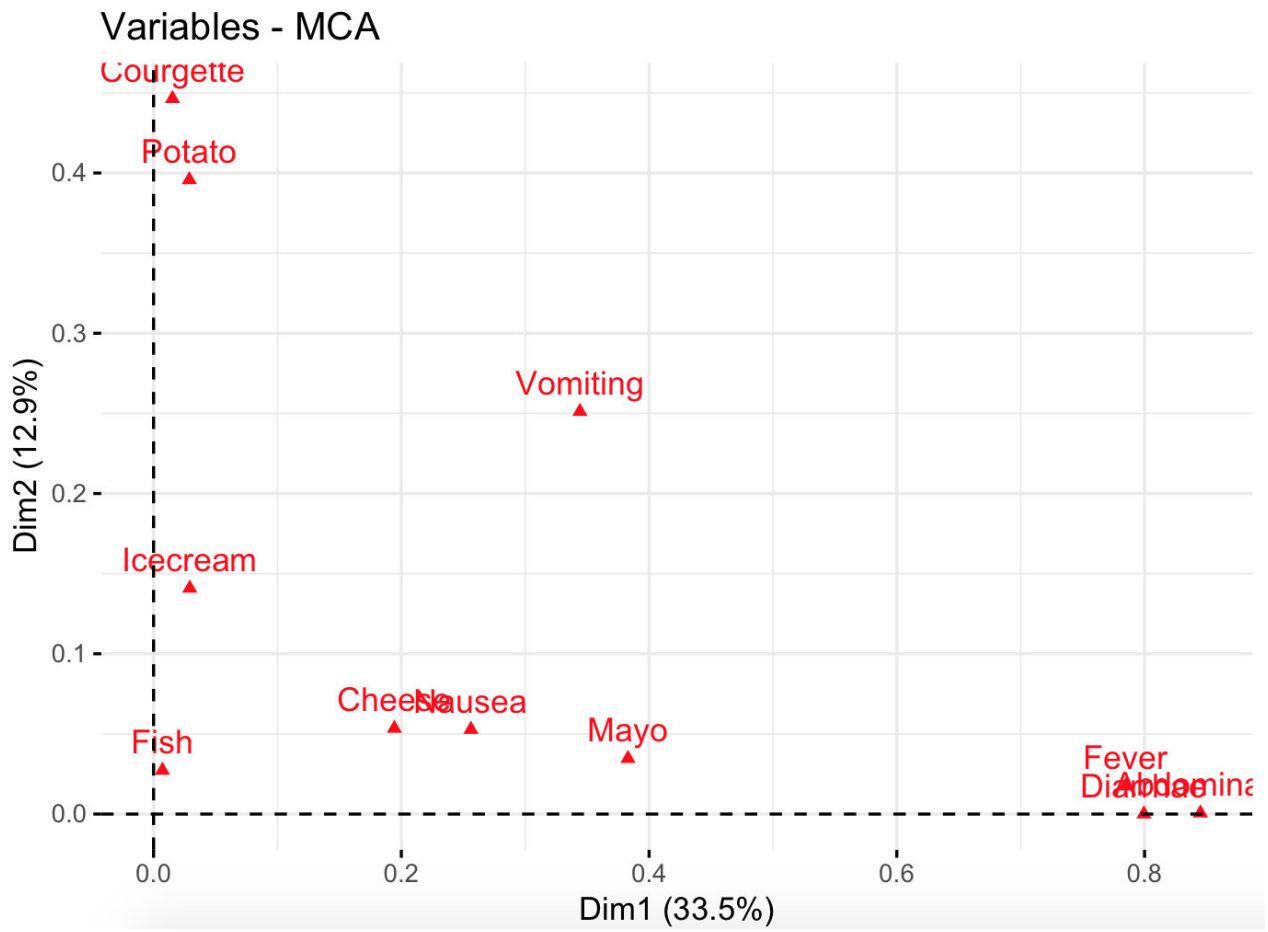
var$coord，对应categories/individuals的坐标

var$cos2,，对应categories/individuals于对应axis(dimension)的相关程度

var$contrib，对应categories/individuals的贡献程度

展示variables和principal dimensions之间的相关度

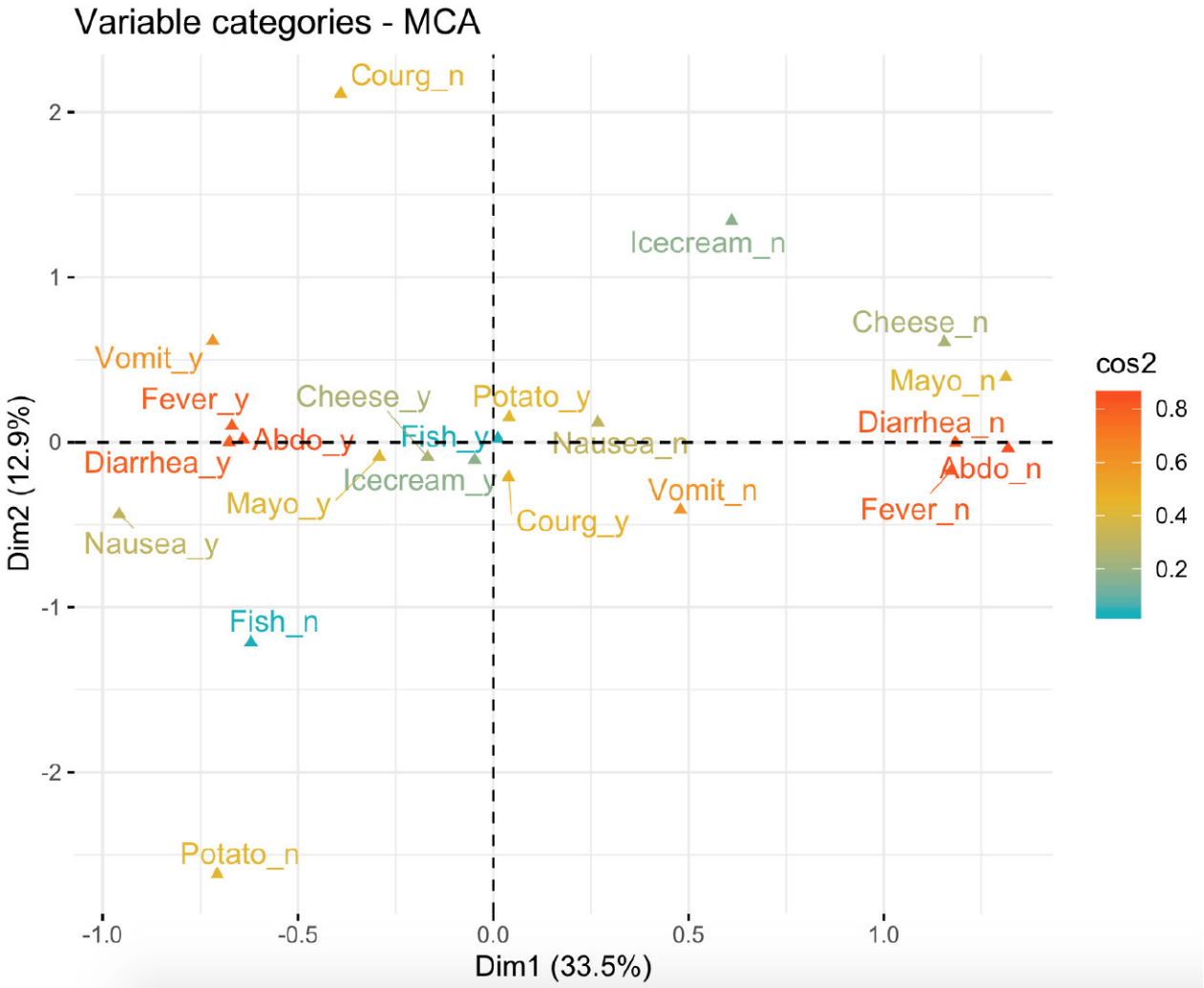
```
fviz_mca_var(res.mca, choice="mca.cor", repel=T)
```



上图帮助识别每个维度最相关的variables，variables这dimensions之间的相关性的平方用于构建坐标位置，可见diarrhae，abdominals和fever和dimension 1非常相关。

展示variables分类坐标位置，同时根据cos2(squared cosine, the quality of the representation)着色绘图

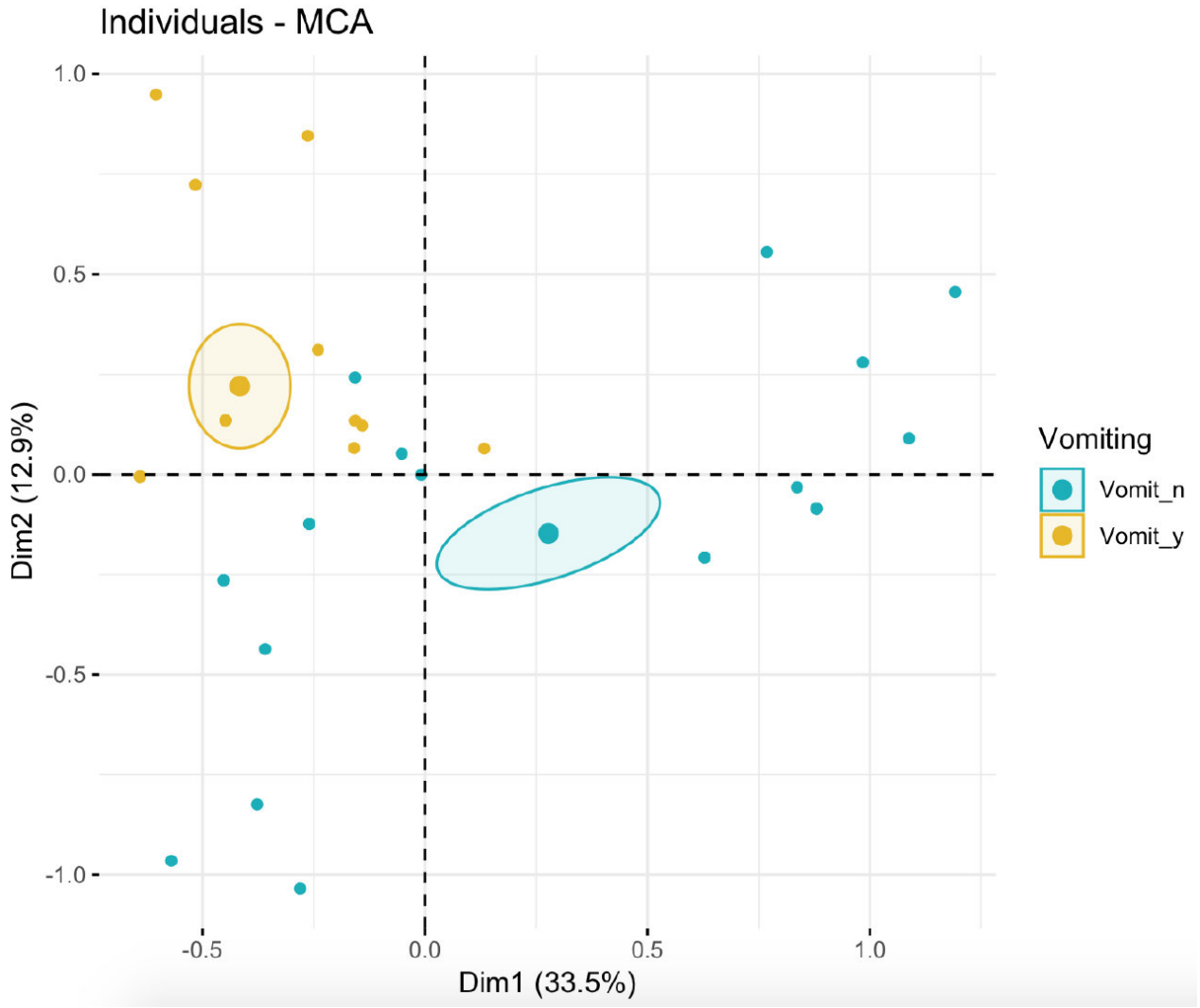
```
fviz_mca_var(res.mca, choice="var.cat", col.var="cos2",
gradient.cols=c("#00AFBB", "#E7B800", "#FC4E07"), repel=T)
```



如图，类似特性的变量分类聚集到一起；负相关的变量分类出现在相对的象限；不同类别点之间的距离和原点对应quality of the variable category on the factor map，远离原点的变量分类更能很好代表factor map

根据individuals值的分类绘制individuals着色图，分组水平为Vomiting，habillage用于指定不同类别individual的颜色

```
fviz_mca_ind(res.mca, label="none", habillage="Vomiting", palette=c("#00AFBB", "#E7B800"), addEllipses=T)
```



或者针对individuals使用多个categories variables绘图

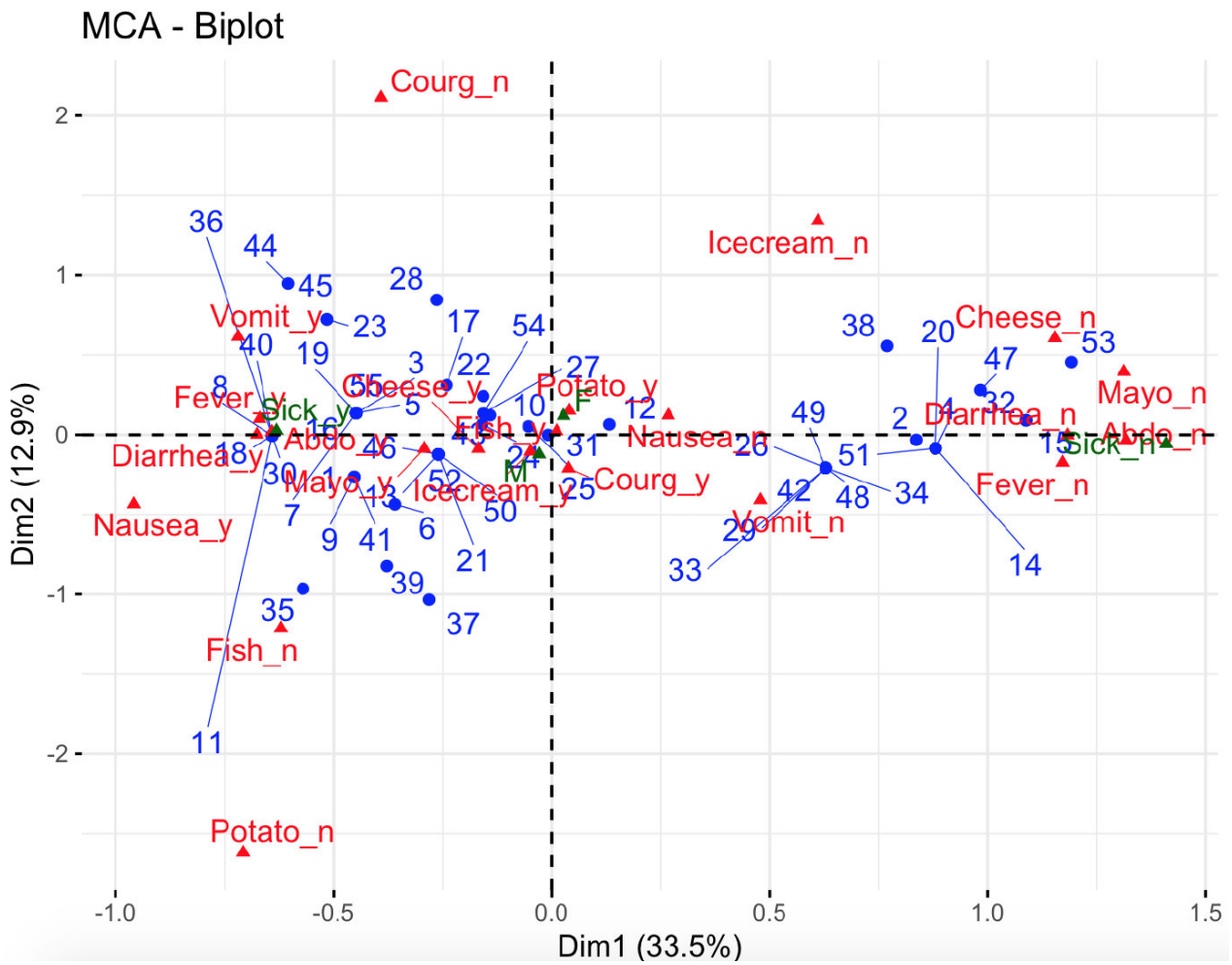
```
fviz_ellipses(res.mca, c("Vomiting", "Fever"), geom="point")
```

或者：`fviz_ellipses(res.mca, 1:4, geom="point")`

MCA，指定quantitative(定量)和qualitative(定型) variables

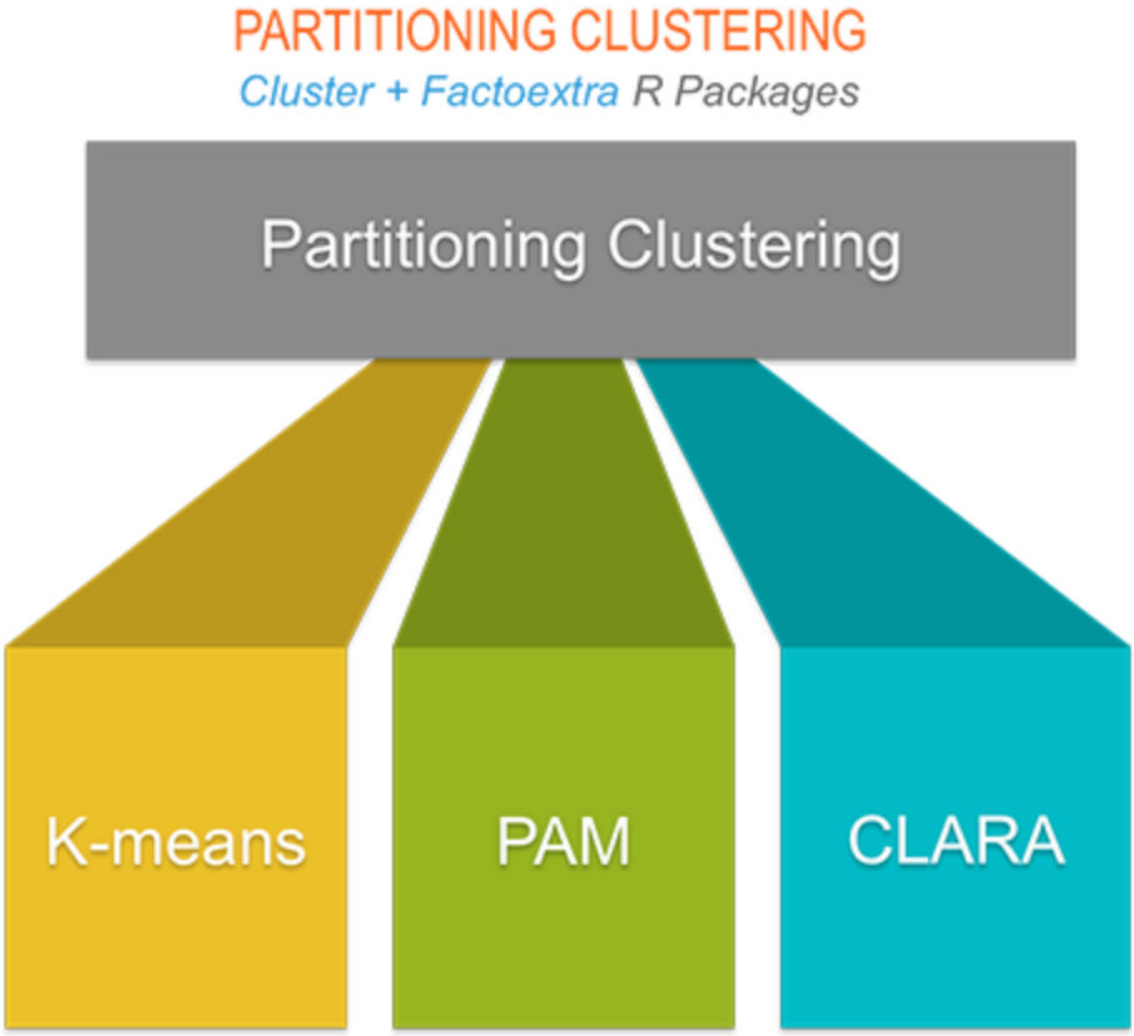
```
res.mca <- MCA(posion, quanti.sup=1:2, quali.sup=3:4, graph=F)
```

```
fviz_mca_biplot(res.mca, repel=TRUE)
```



如图，individuals蓝色，supplyment individuals深蓝色，variables红色，supplyment variables深红色。

Partitioning clustering



kmeans聚类分析

```
data(USArrests)
```

```
df <- scale(USArrests)
```

```
km.res <- kmeans(scale(USArrest), 4, nstart=25)
```

首先针对每列进行标准化处理，防止出现差异较大导致无法正常显示；其次kmeans函数计算每列数值的聚类分布，根据参数center=4，将每列分为4类；nstart=25，选择25个随机点开始聚类。

其聚类过程为：首先分别针对所有列进行指定数目的聚类分布，可得，对应4类的中心位置：

```
> km.res$centers
      Murder    Assault   UrbanPop        Rape
1  1.4118898  0.8743346 -0.8145211  0.01927104
2 -0.4894375 -0.3826001  0.5758298 -0.26165379
3 -0.9615407 -1.1066010 -0.9301069 -0.96676331
4  0.6950701  1.0394414  0.7226370  1.27693964
```

然后计算每一行到这4类中心的距离，距离最短的，及属于该类：

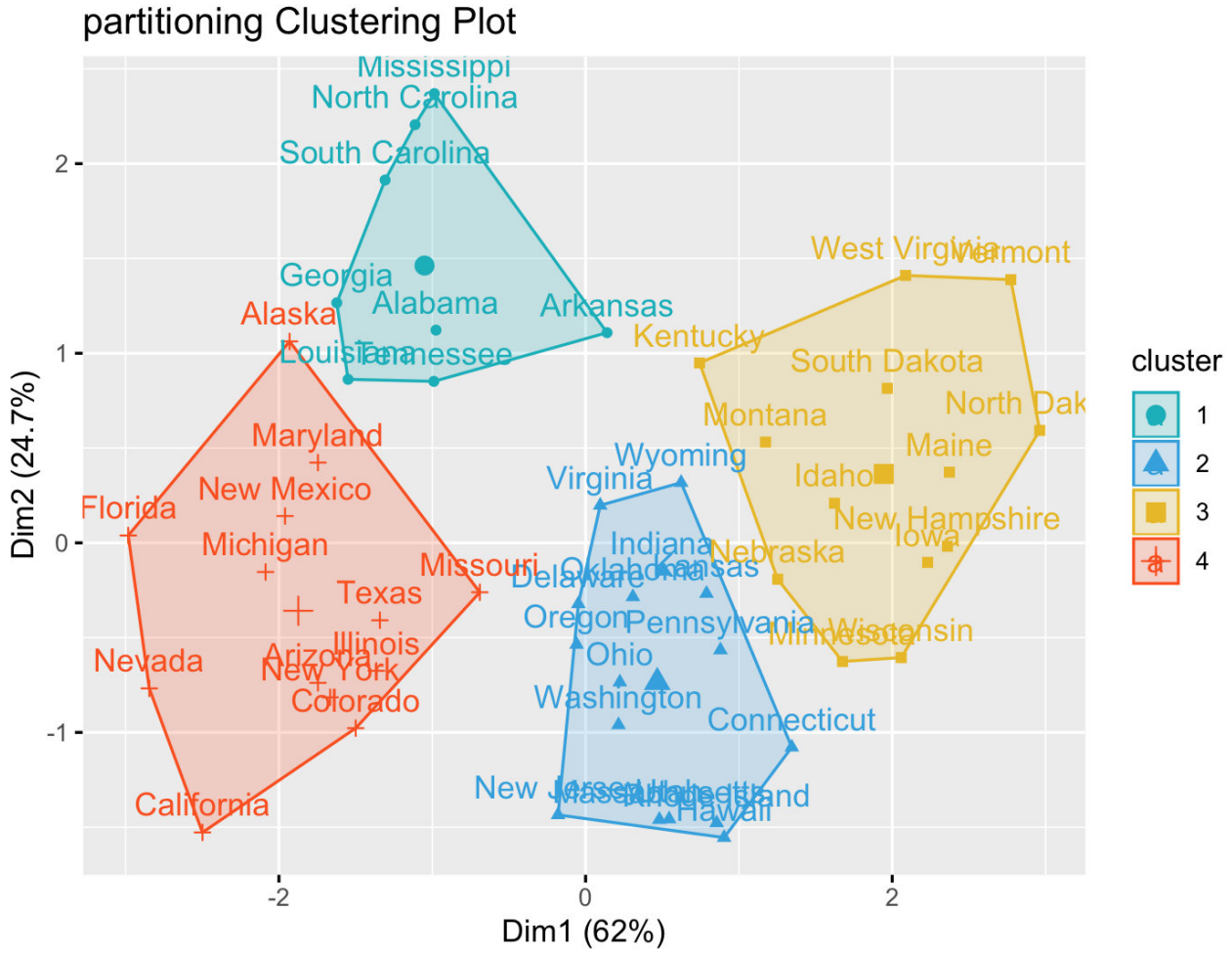
```
> head(df)
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
```

例如 Alabama

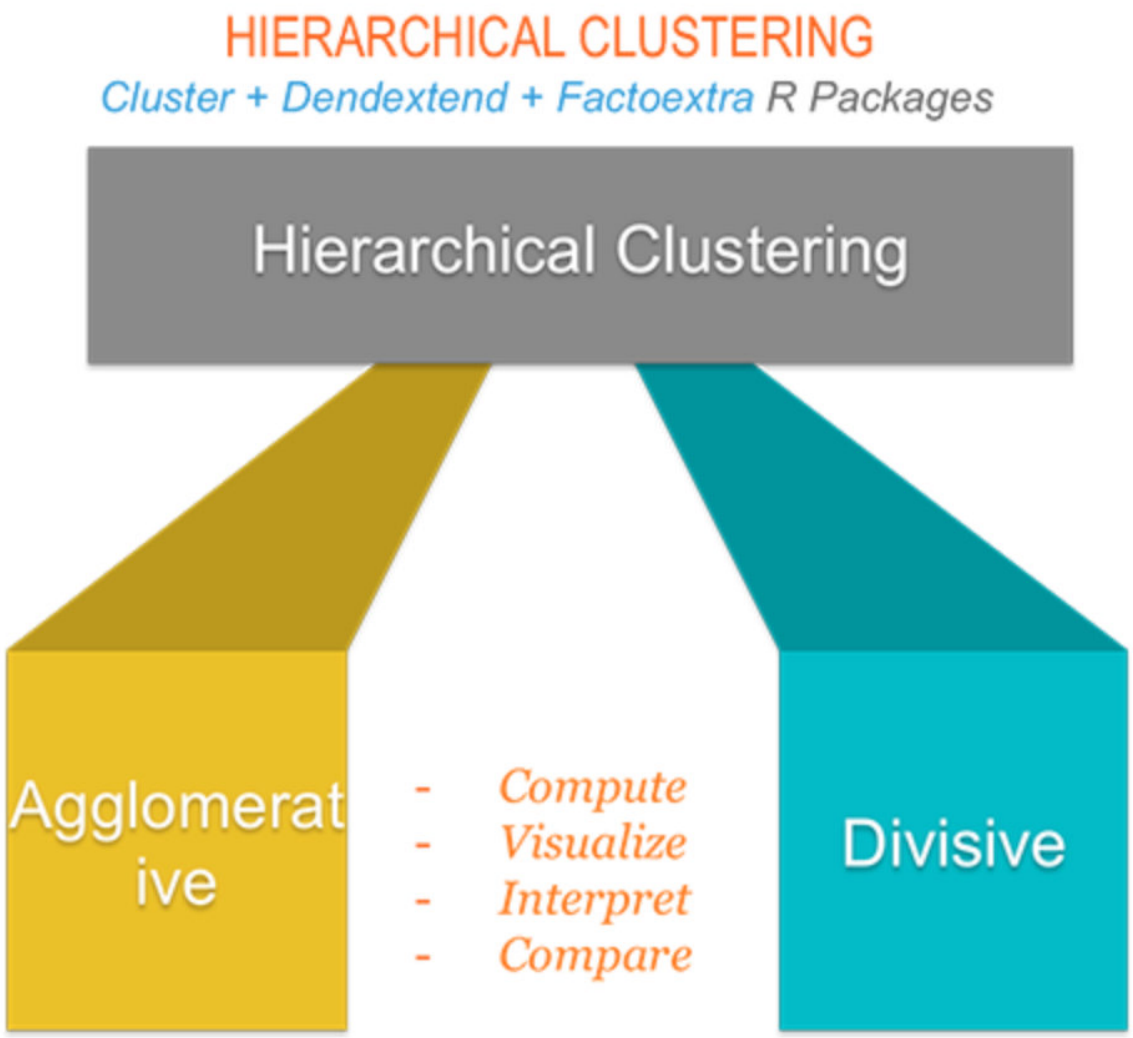
```
> for(i in 1:4){
+ value <- (df[1,] - km.res$centers[i,])^2
+ value <- sum(value)
+ print(value)}
[1] 0.1237668
[1] 5.627596
[1] 9.523545
[1] 3.551307
```

绘制聚类图

```
fviz_cluster(km.res, data=df, palette=c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"), main="Partitioning Clustering Plot")
```



Hierarchical clustering



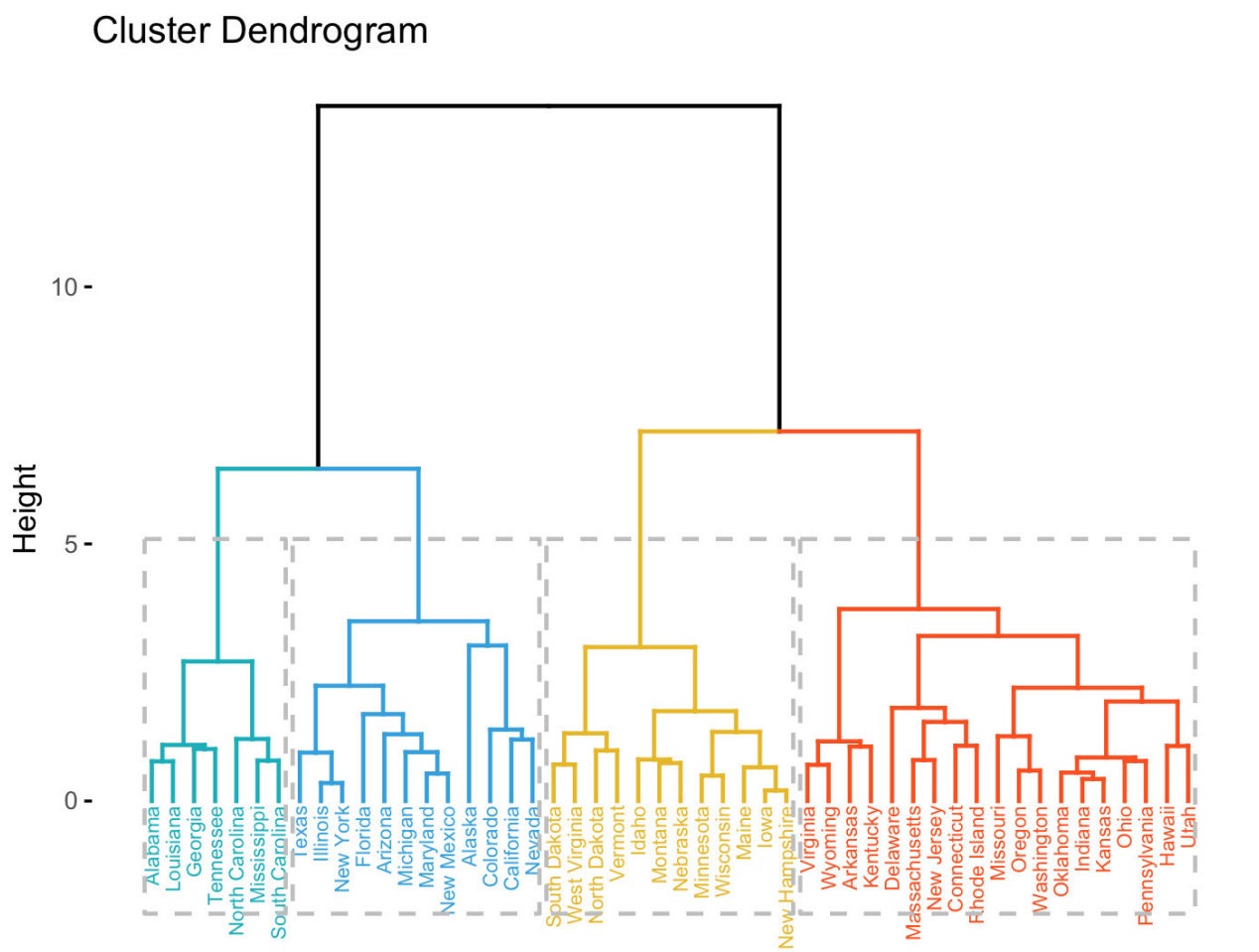

```
library(factoextra)
```

hcut：计算分级聚类并将系统发育树非常对应类别数目。计算分级聚类函数可选hc_func=hclust(默认), agnes, diana；同时可以根据基于hc_metric=person，spearman，kendall计算的相关性聚类。

```
res <- hcut(USArrests, k=4, stand=T)
```

k=4指定聚类数目，stand=T，scale 每列数据

```
fviz_dend(res, rect=T, cex=0.5, k_colors=c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"))
```



Determine the optimal number of clusters

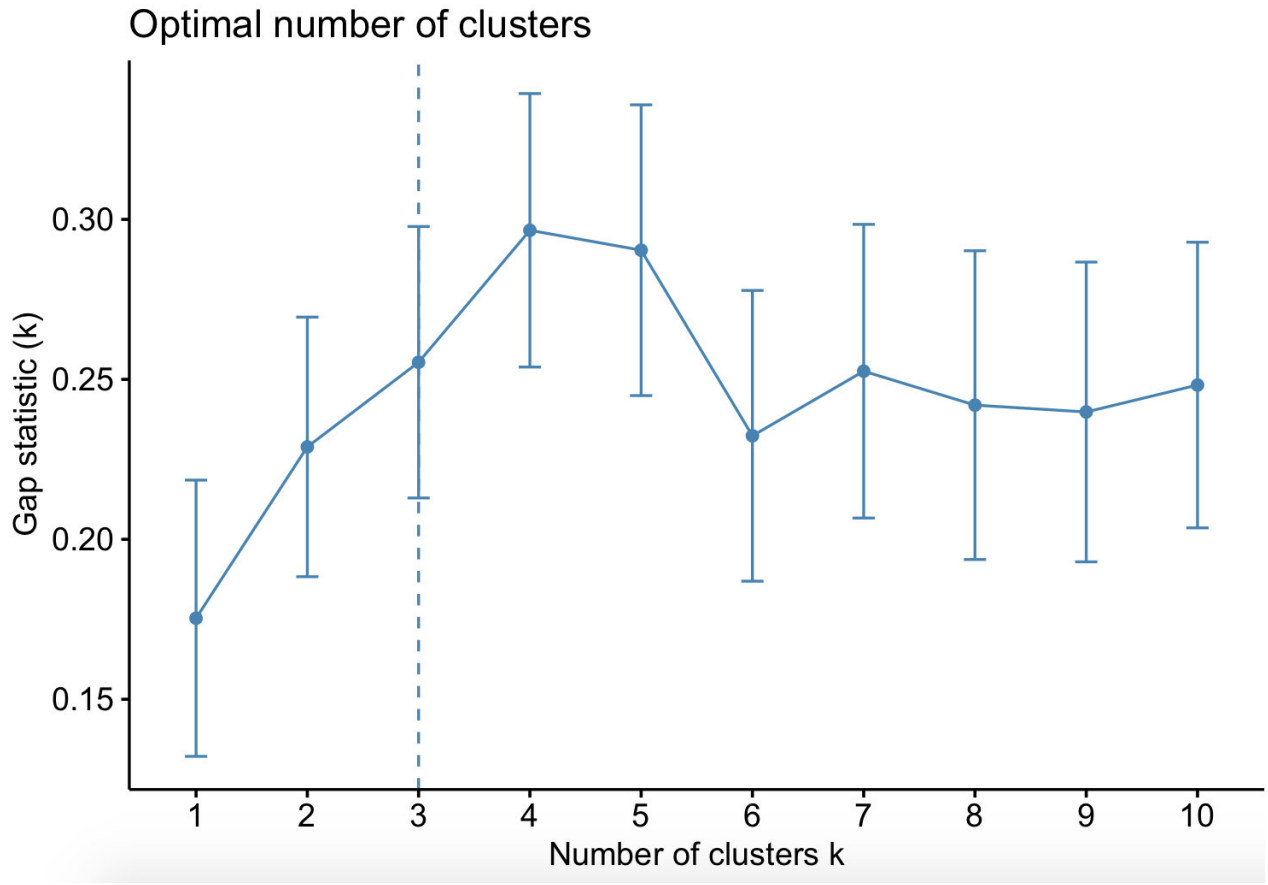
fviz_nbclust函数用于确定聚类个数

```
library(factoextra)
```

```
my_data <- scale(USArrests)
```

```
fviz_nbclust(my_data, FUNcluster=kmeans, method="gap_stat")
```

FUNcluster，指定采用的聚类方法,Allowed values include: kmeans, cluster::pam, cluster::clara, cluster::fanny, hcut；method，评估最佳聚类个数的方法，Possible values are "silhouette" (for average silhouette width), "wss" (for total within sum of square) and "gap_stat" (for gap statistics)



Gap statistic for hierarchical clustering

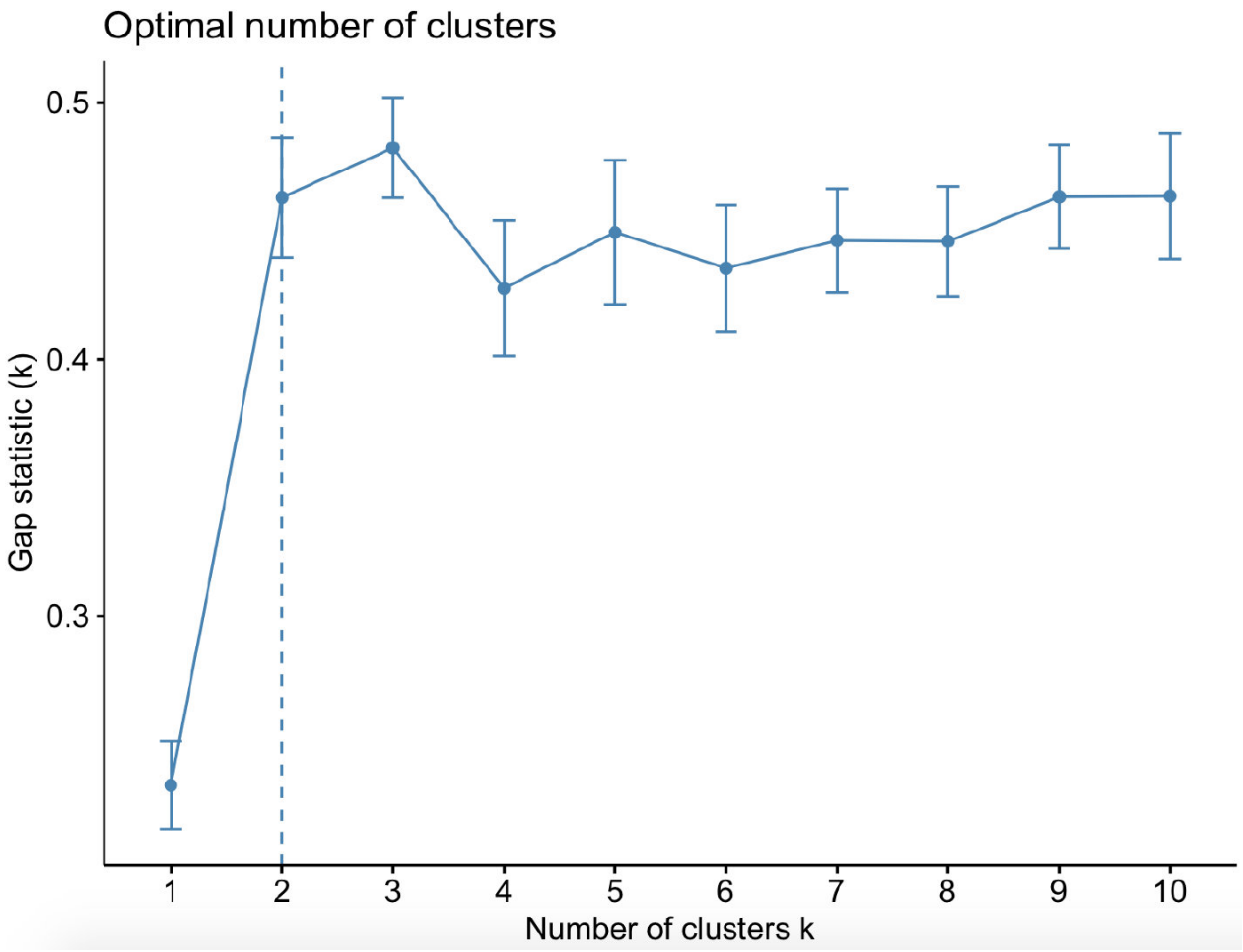
```
data(iris)
```

```
iris.scaled <- scale(iris[,-5])
```

```
gat_stat <- clusGap(iris.scaled, FUN=kmeans, nstart=25, k.max=10,B=10)
```

nstart=25，指定随机数据集数目；k.max，最大的聚类数目；B，bootstrap样本数目

```
fviz_gap_stat(gap_stat)
```




```
gap_stat <- clusGap(iris.scaled, FUN = hcut, K.max = 10, B = 10)
```

```
fviz_gap_stat(gap_stat)
```