Cluster Analysis(CA) and PCA(Principal Component Analysis)

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(neuralnet)
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(rpart)
library(rattle)
```

```
## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(MASS)
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine()       masks randomForest::combine()
## x dplyr::compute()       masks neuralnet::compute()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::lift()          masks caret::lift()
## x randomForest::margin() masks ggplot2::margin()
## x dplyr::select()        masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## The following object is masked from 'package:bitops':
##
##     %&%
##
## Loaded glmnet 4.1-8
```

```
library(leaps)
library(ggplot2)
```

1.

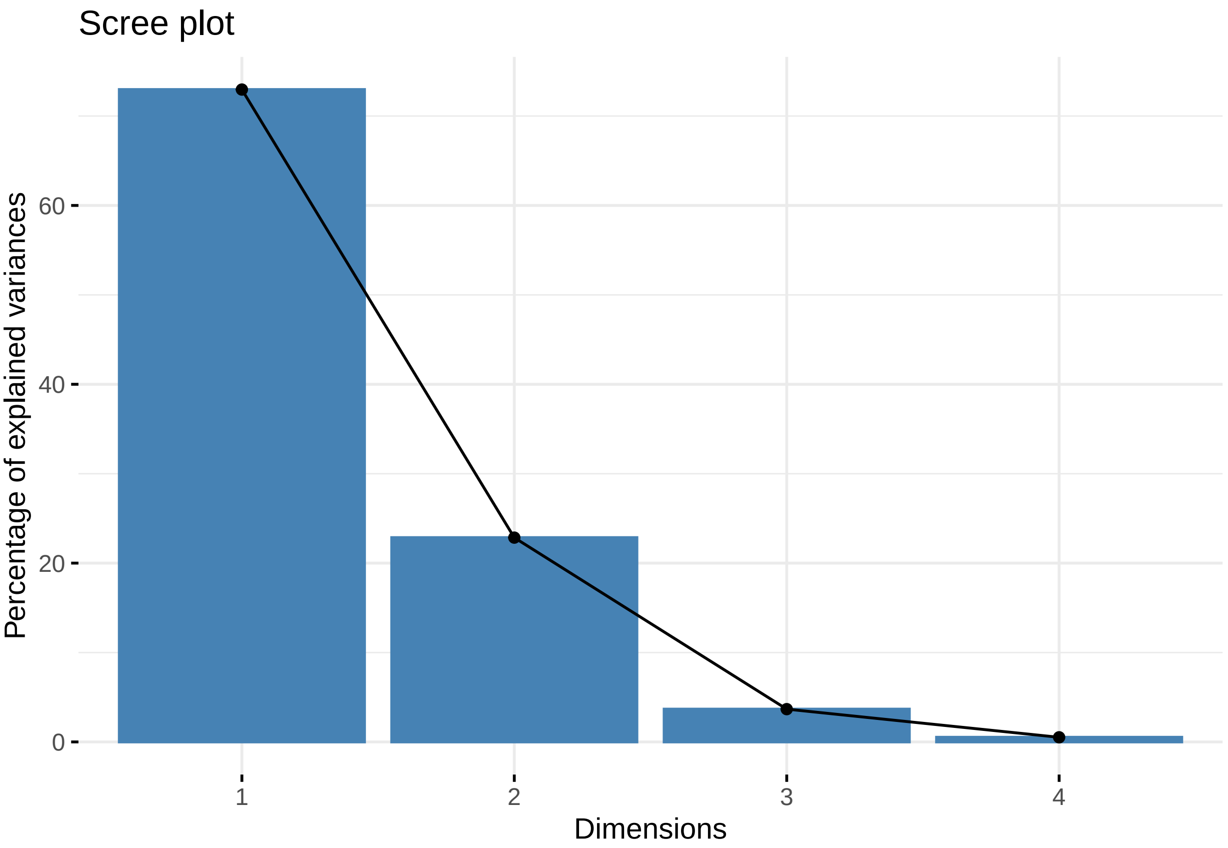
```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

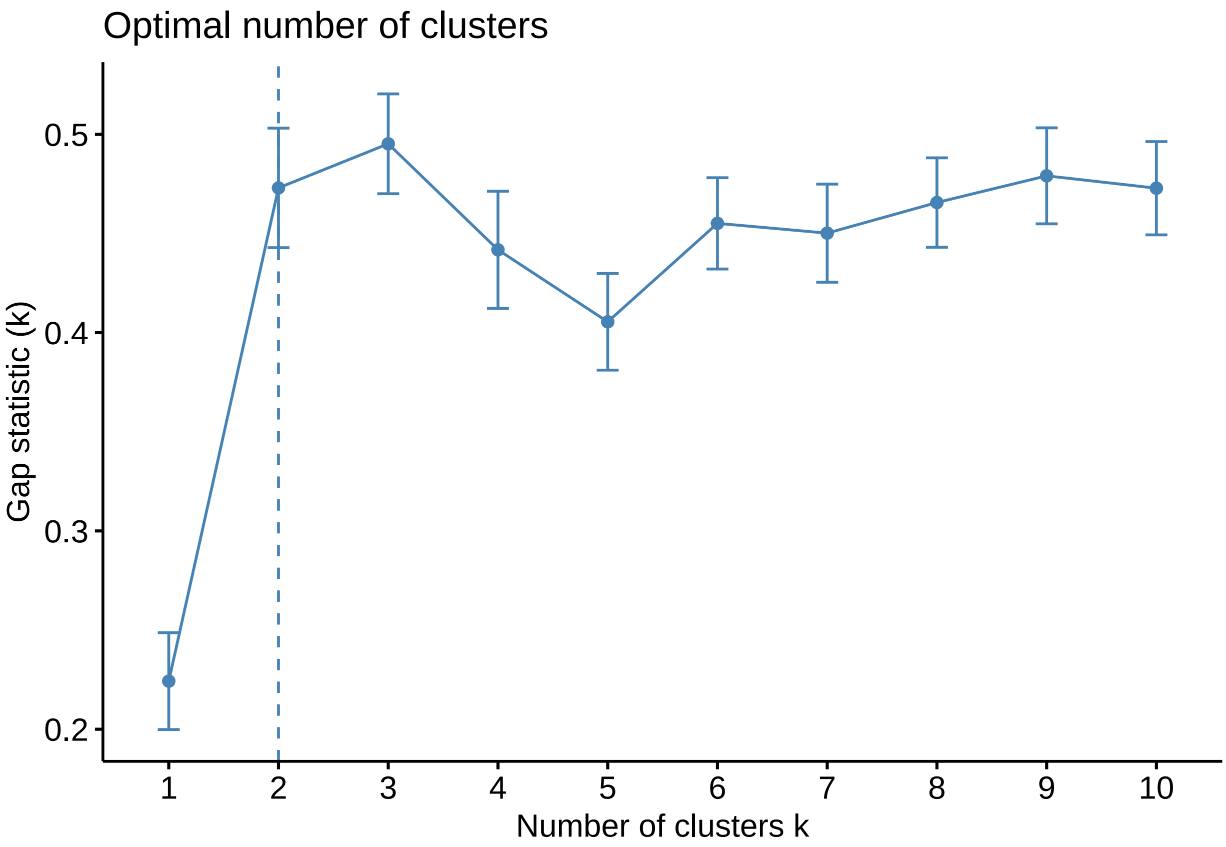
```
data <- na.omit(iris)
data <- scale(data[,-5])
pc <- princomp(data,cor=TRUE)
# princomp performs a principal components analysis on the given numeric data matrix and returns the re

# Scree-plot
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(pc)
```

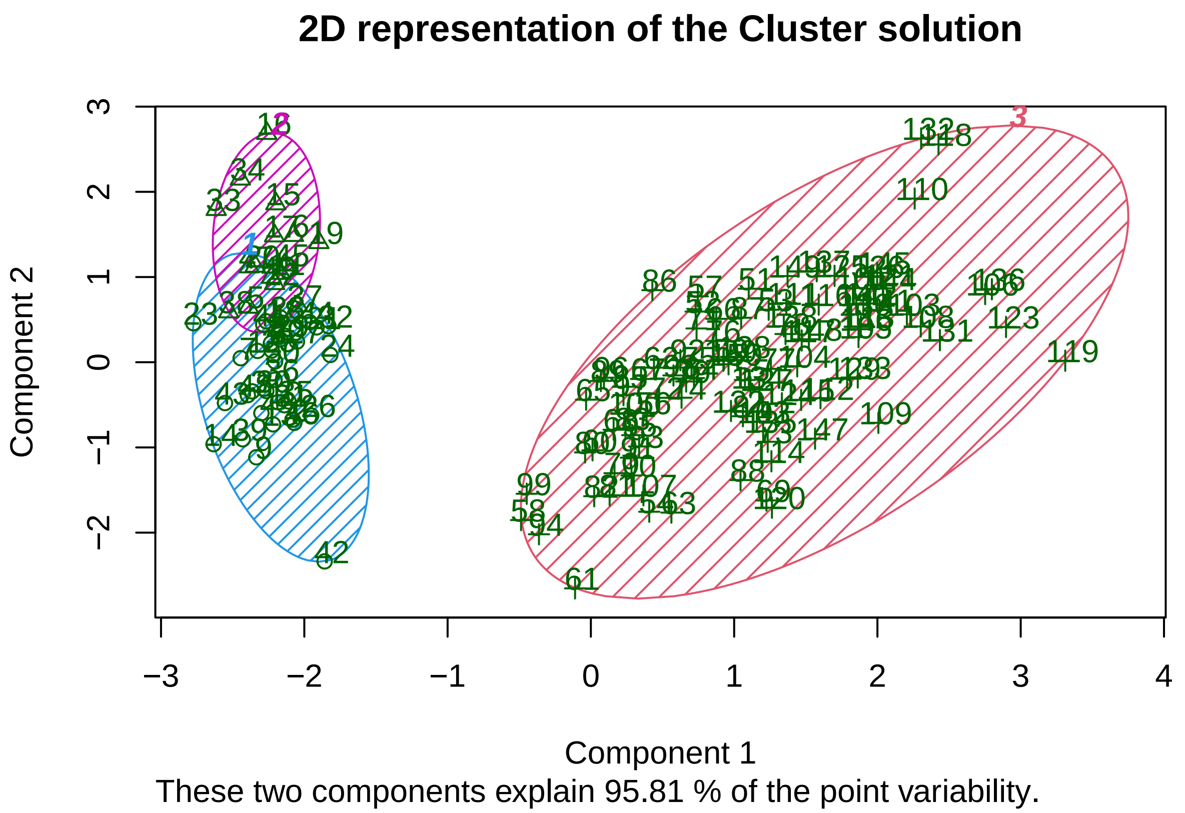


```
fviz_nbclust(data,kmeans,method="gap_stat")
```



K means

```
k.means.fit <- kmeans(data,3)
library(cluster)
clusplot(data,k.means.fit$cluster,main="2D representation of the Cluster solution",color=TRUE,shade=TRU
```



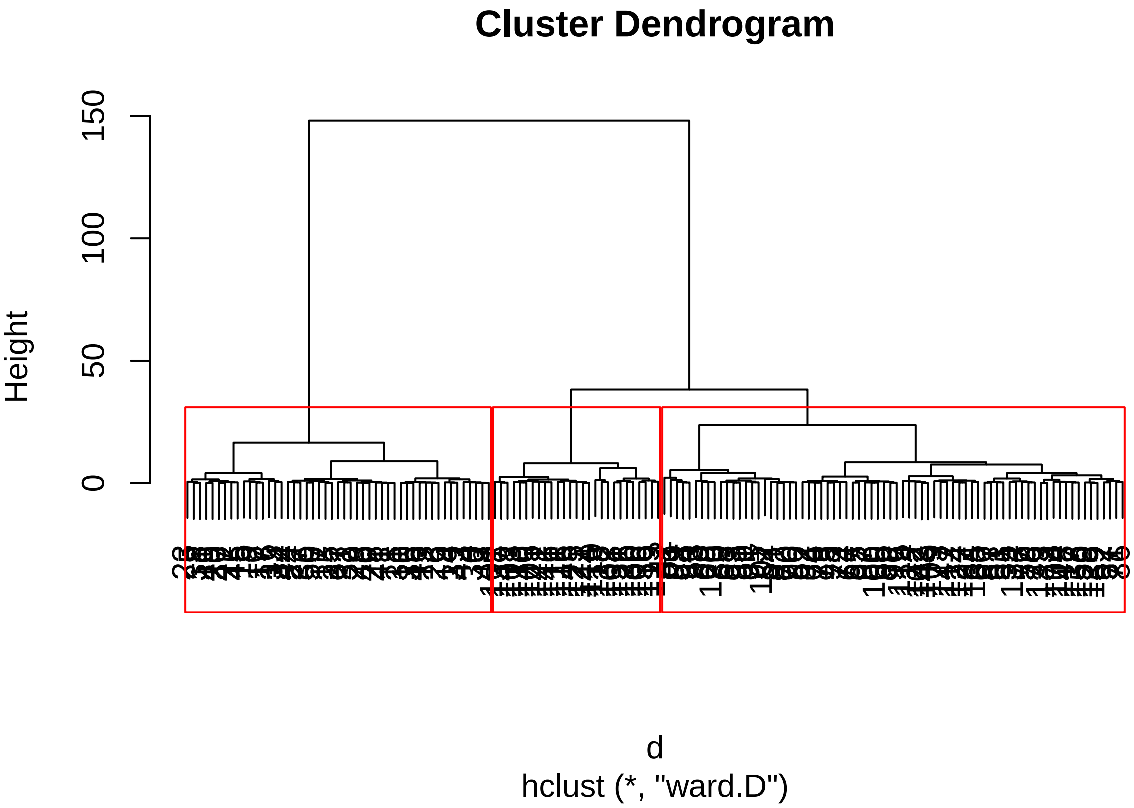
```
table(k.means.fit$cluster,iris$Species)
```

```
##
##     setosa versicolor virginica
##   1     34          0         0
##   2     16          0         0
##   3      0         50        50
```

```
# accuracy = (50+39+36)/150 = 125/150
```

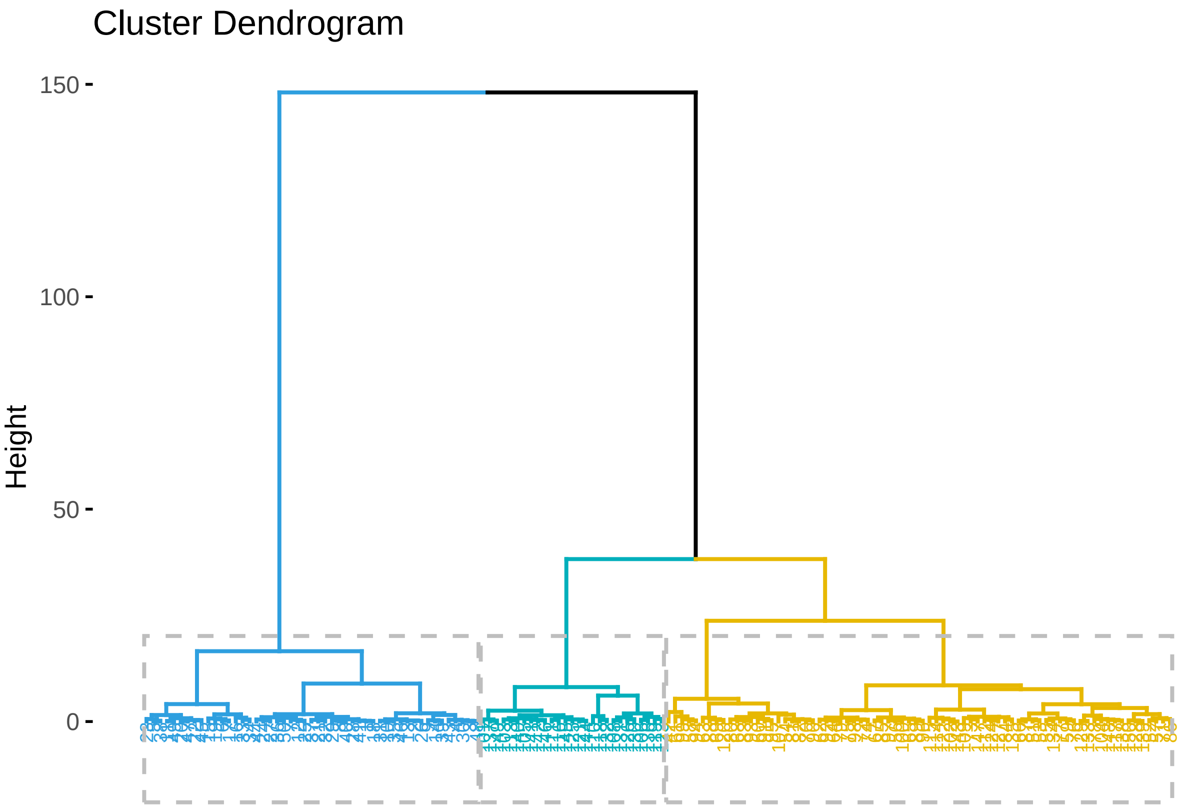
H.ward

```
d <- dist(data, method="euclidean")
H.fit <- hclust(d,method="ward.D")
plot(H.fit)
rect.hclust(H.fit,k=3,border="red")
```



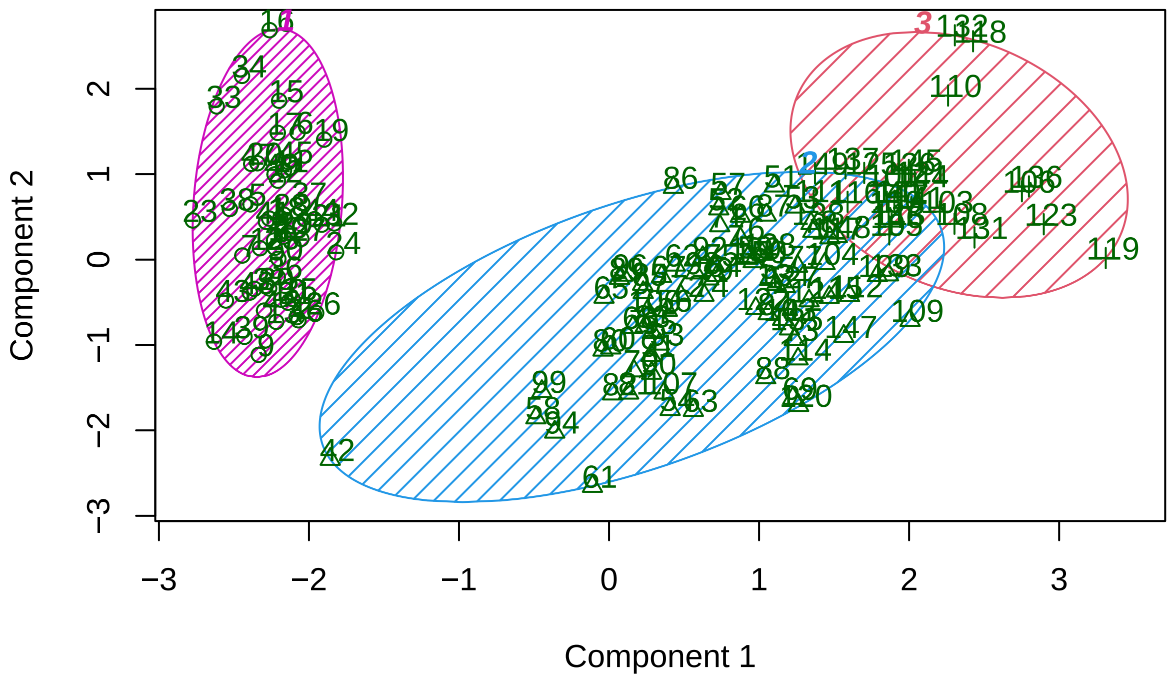
```
fviz_dend(H.fit,k=3,cex=0.5,k_colors=c("#2E9FDF","#00AFBB","#E7B800"),color_labels_by_k = TRUE,rect = T
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
groups <- cutree(H.fit,k=3)
clusters <- factor(groups, levels=1:3,labels=c("setosa","versicolor","virginica"))
clusplot(data,groups,main="2D representation of the Cluster solution", color=TRUE, shade=TRUE, labels=2
```

2D representation of the Cluster solution



These two components explain 95.81 % of the point variability.

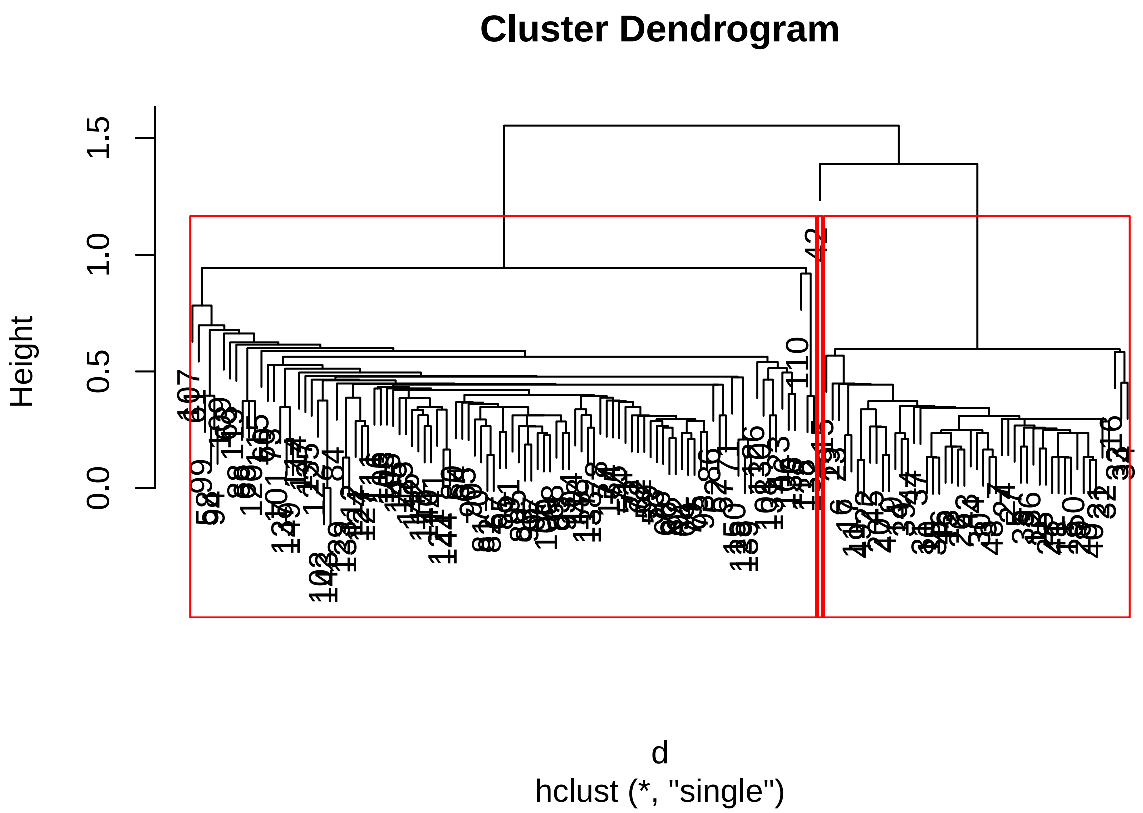
```
table(iris[,5],clusters)
```

```
##             clusters
##              setosa versicolor virginica
##   setosa         49          1         0
##   versicolor      0         50         0
##   virginica       0         23        27
```

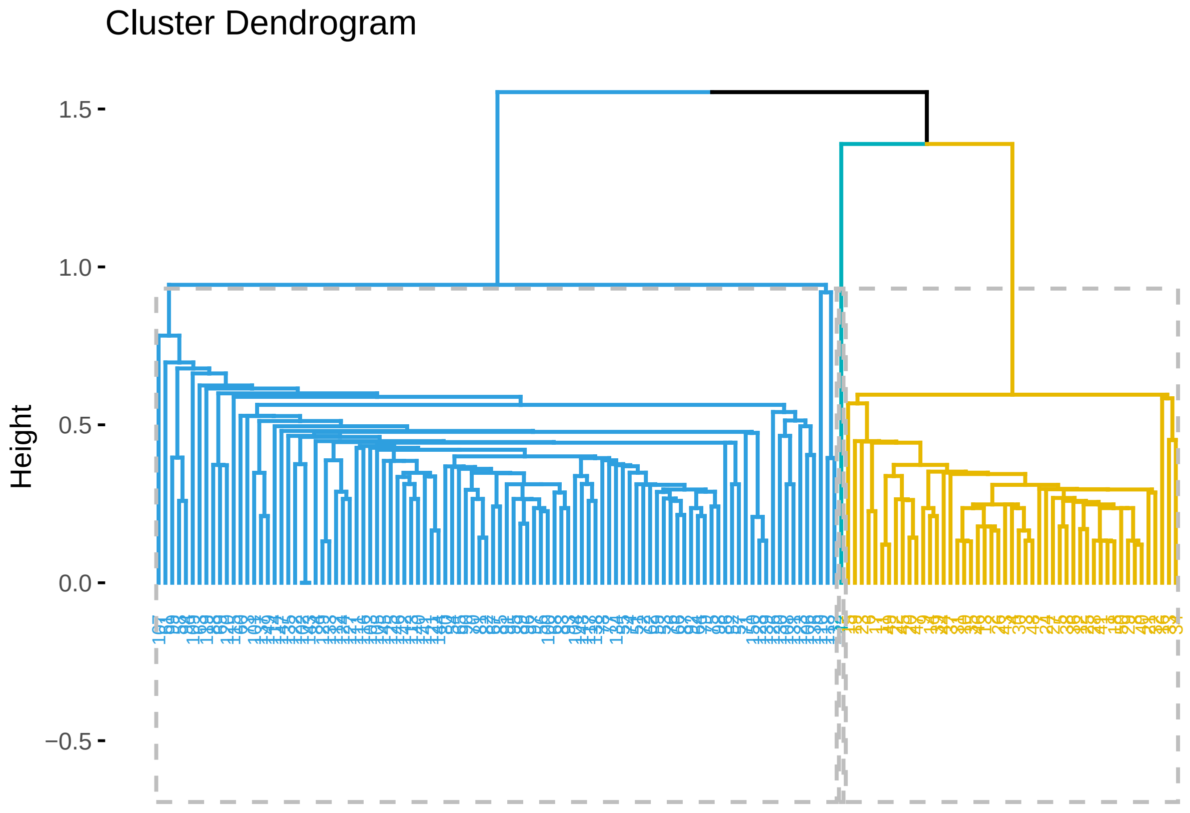
```
# accuracy = (49+50+27)/150
```

H.single

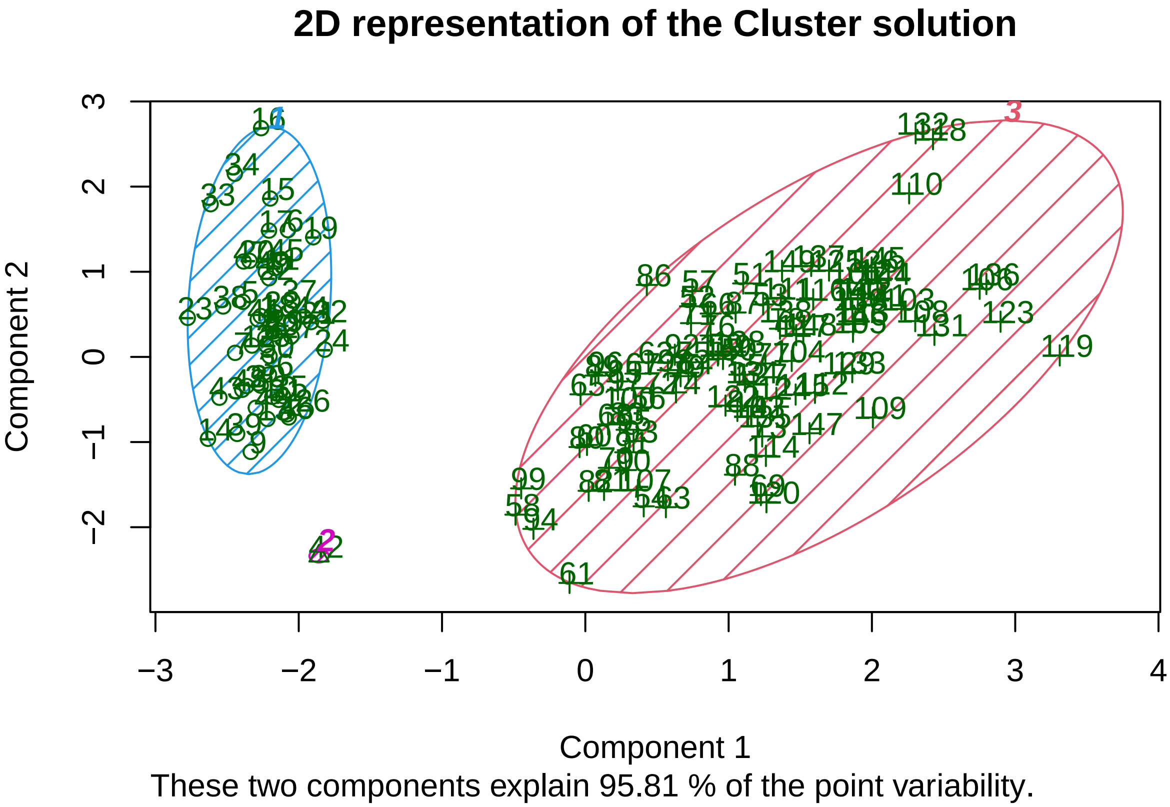
```
H.fit <- hclust(d,method="single")
plot(H.fit)
rect.hclust(H.fit,k=3,border="red")
```



```
fviz_dend(H.fit,k=3,cex=0.5,k_colors=c("#2E9FDF","#00AFBB","#E7B800"),color_labels_by_k = TRUE,rect = T
```



```
groups <- cutree(H.fit,k=3)
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE, shade=TRUE, labels =
```



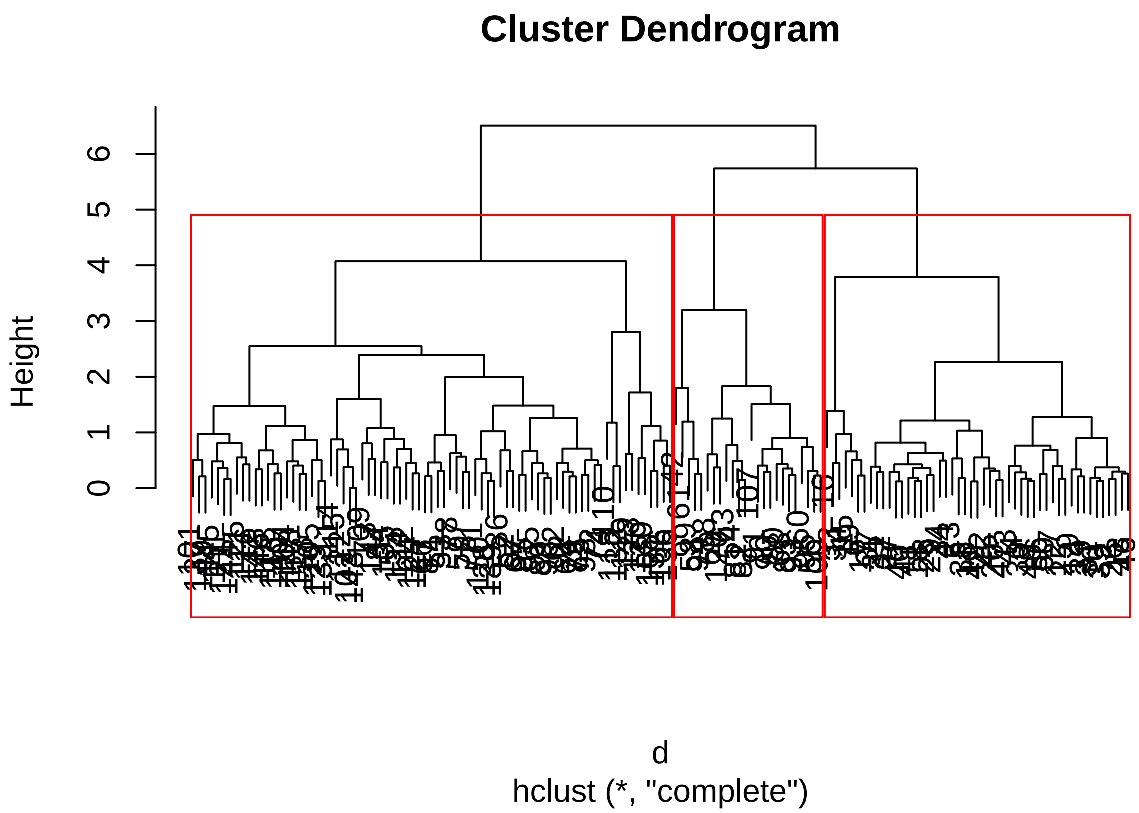
```
clusters <- factor(groups,levels=1:3,labels=c("setosa","versicolor","virginica"))
table(iris[,5],clusters)
```

```
##             clusters
##              setosa versicolor virginica
##   setosa         49          1         0
##   versicolor      0          0        50
##   virginica       0          0        50
```

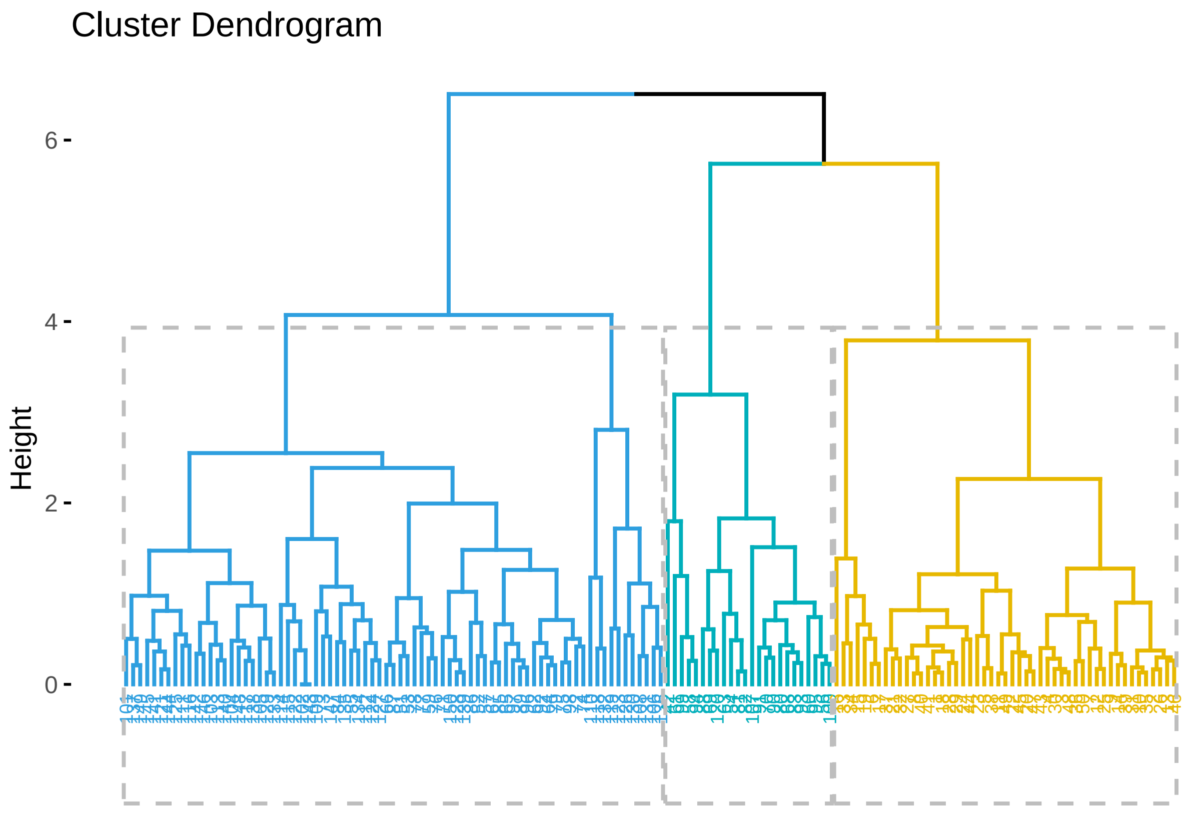
```
# accuracy = (49+50)/150
```

H.complete

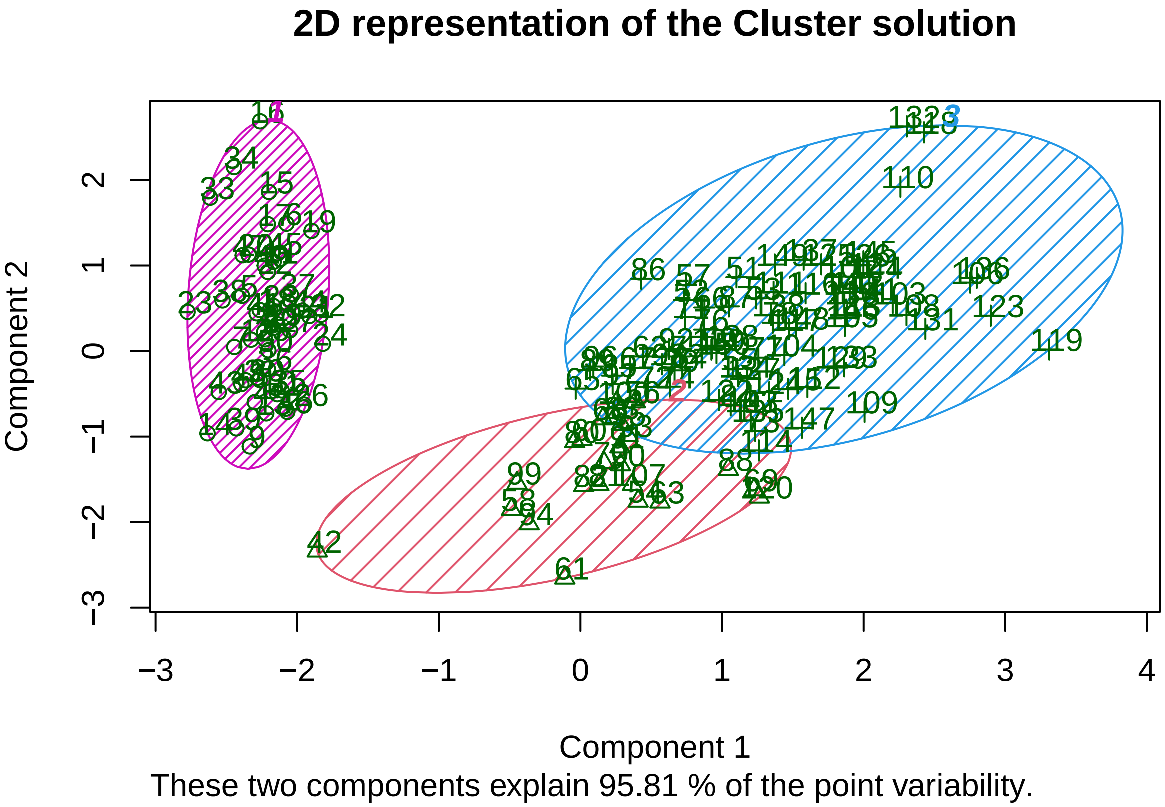
```
H.fit <- hclust(d, method="complete")
plot(H.fit)
rect.hclust(H.fit,k=3,border="red")
```



```
fviz_dend(H.fit,k=3,cex=0.5,k_colors=c("#2E9FDF","#00AFBB","#E7B800"),color_labels_by_k = TRUE,rect = T
```



```
groups <- cutree(H.fit,k=3)
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE,shade=TRUE,labels=2,li
```



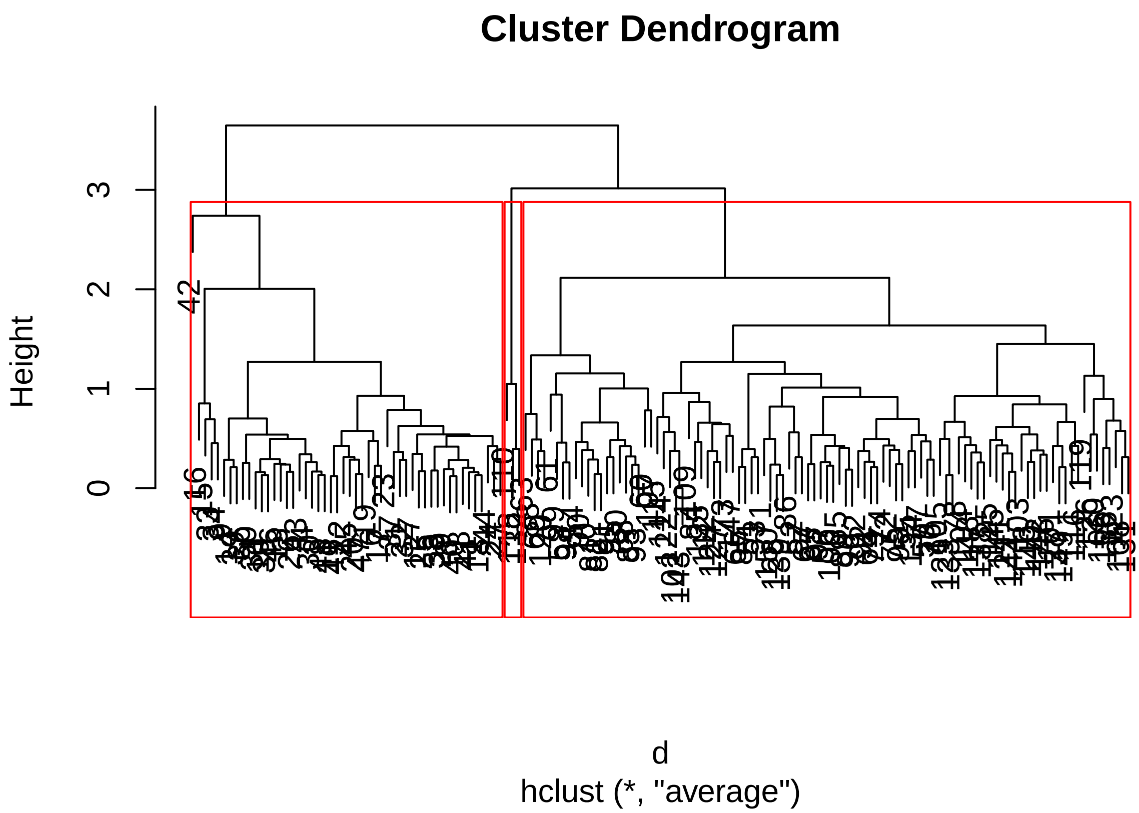
```
clusters <- factor(groups,levels=1:3,labels=c("setosa","versicolor","virginica"))
table(iris[,5],clusters)
```

```
##              clusters
##                setosa versicolor virginica
##    setosa          49          1         0
##    versicolor       0         21        29
##    virginica        0          2        48
```

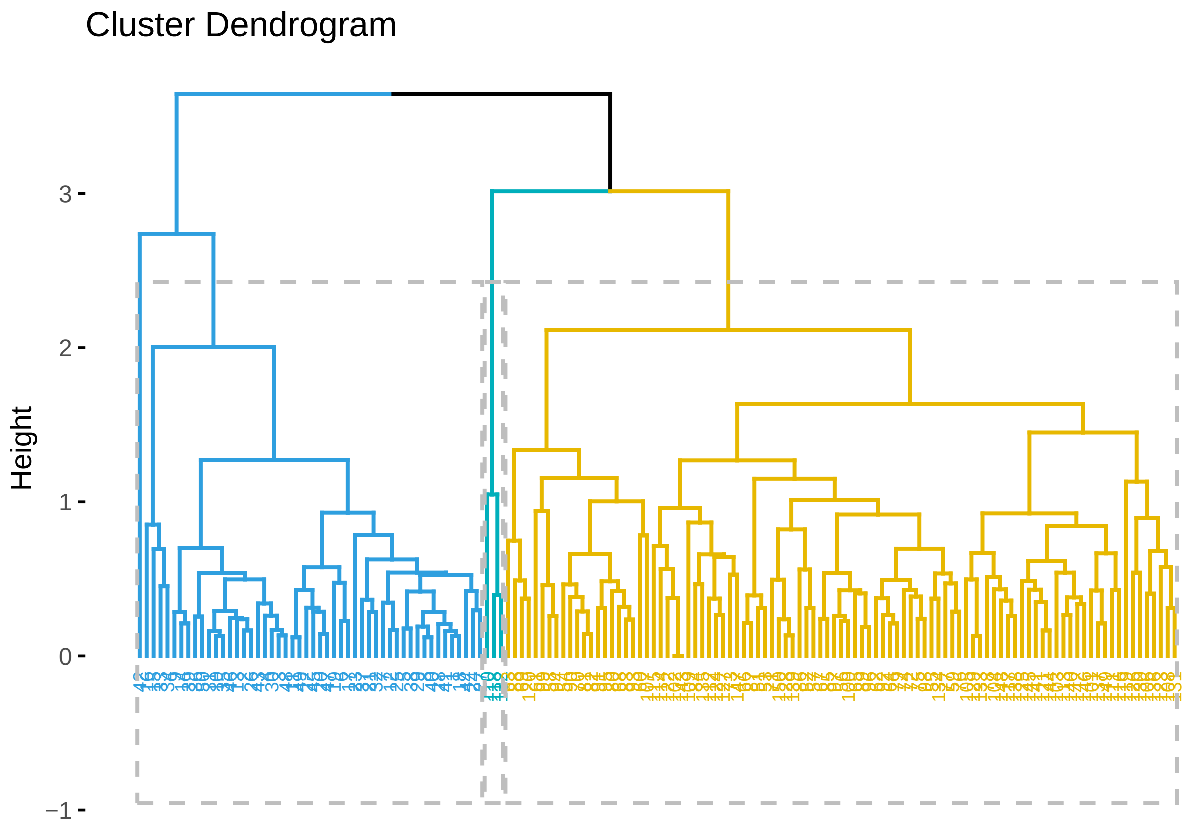
```
# accuracy = (49+21+48)/150
```

H.Average

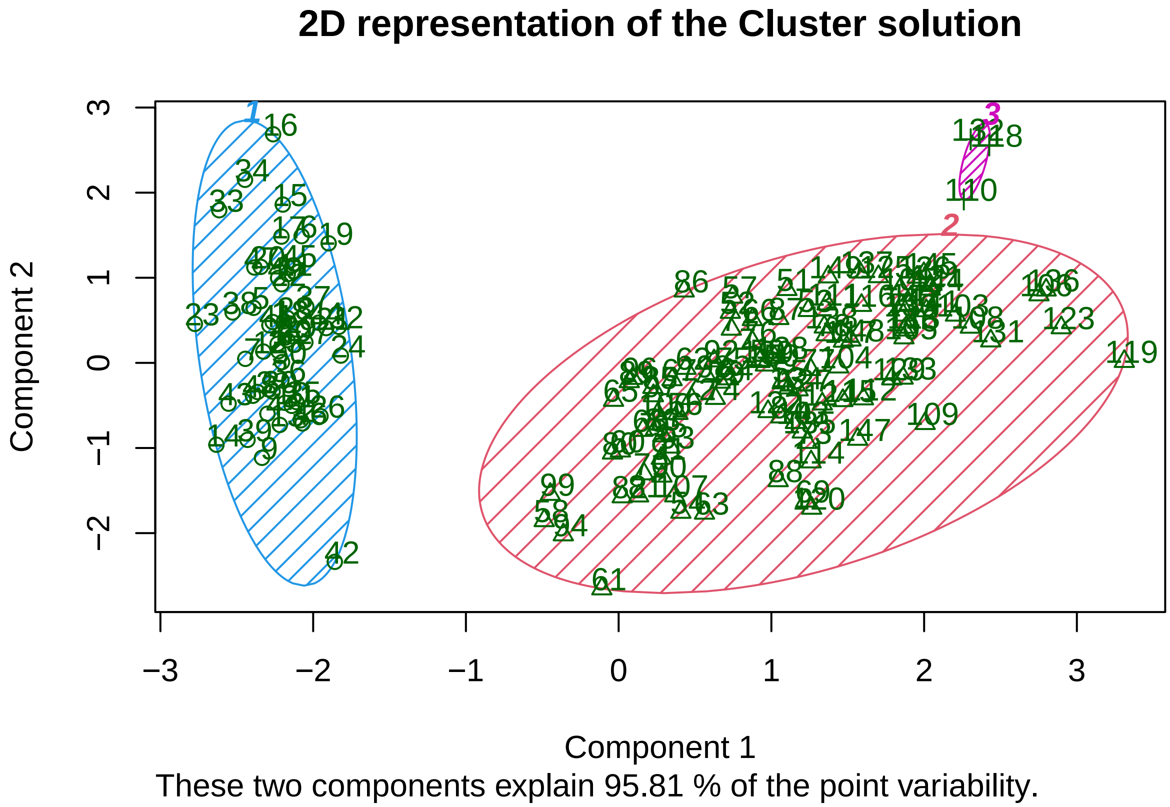
```
H.fit <- hclust(d,method="average")
plot(H.fit)
rect.hclust(H.fit,k=3,border="red")
```



```
fviz_dend(H.fit,k=3,cex=0.5,k_colors=c("#2E9FDF","#00AFBB","#E7B800"),color_labels_by_k = TRUE,rect = T
```



```
groups <- cutree(H.fit,k=3)
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE,shade=TRUE,labels = 2,
```



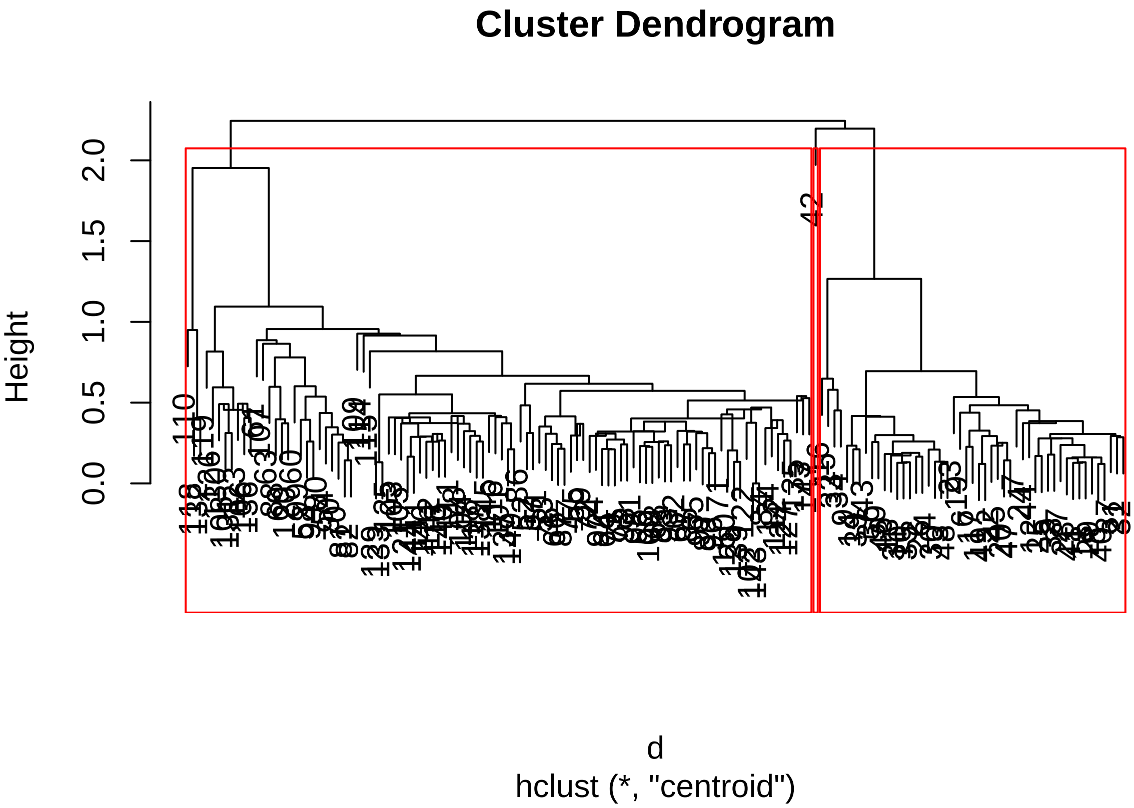
```
clusters <- factor(groups,levels=1:3,labels=c("setosa","versicolor","virginica"))
table(iris[,5],clusters)
```

```
##             clusters
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         50         0
##   virginica       0         47         3
```

```
# accuracy = (50+50+3)/150
```

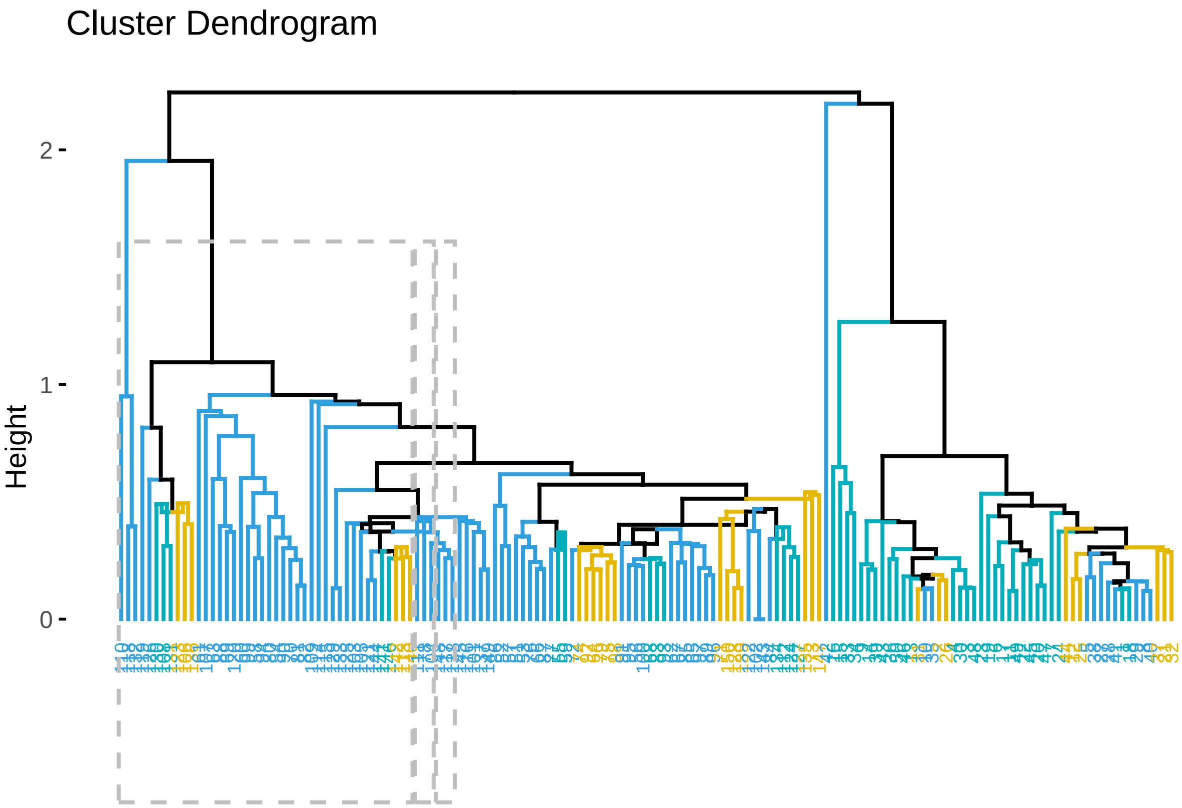
H.centroid

```
H.fit <- hclust(d,method="centroid")
plot(H.fit)
rect.hclust(H.fit,k=3,border="red")
```

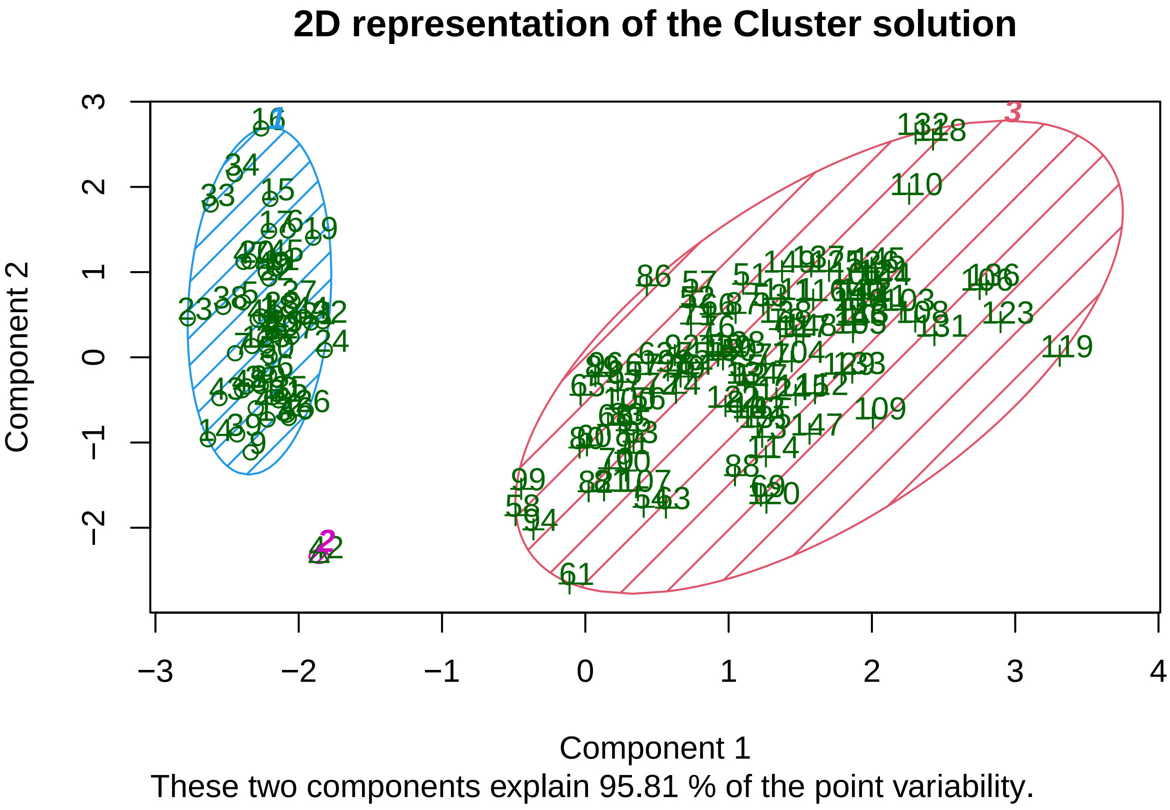


```
fviz_dend(H.fit,k=3,cex=0.5,k_colors=c("#2E9FDF","#00AFBB","#E7B800"),color_labels_by_k = TRUE,rect = T
```

```
## Warning in get_col(col, k): Length of color vector was shorter than the number
## of clusters - color vector was recycled
```



```
groups <- cutree(H.fit,k=3)
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE,shade = TRUE, labels =
```



```
clusters <- factor(groups,levels=1:3,labels=c("setosa","versicolor","virginica"))
table(iris[,5],clusters)
```

```
##               clusters
##                setosa versicolor virginica
##   setosa           49          1         0
##   versicolor        0          0        50
##   virginica         0          0        50
```

```
# accuracy = (49+50)/150
```

Comparision Ward > K-means > Complete > Average > Single = Centroid **I would recommend the cluster analysis as a good way to deal with the IRIS dataset. K-means and complete methods are usually great, but here Ward method is the best one from the confusion matrix.**

2.

```
library(devtools)
```

```
## Loading required package: usethis
```

```
## Warning: package 'usethis' was built under R version 4.2.3
```

```
library(ggbiplot)
```

```
## Warning: package 'ggbiplot' was built under R version 4.2.3
```

```
##
## Attaching package: 'ggbiplot'
```

```
## The following object is masked from 'package:rattle':
##
##     wine
```

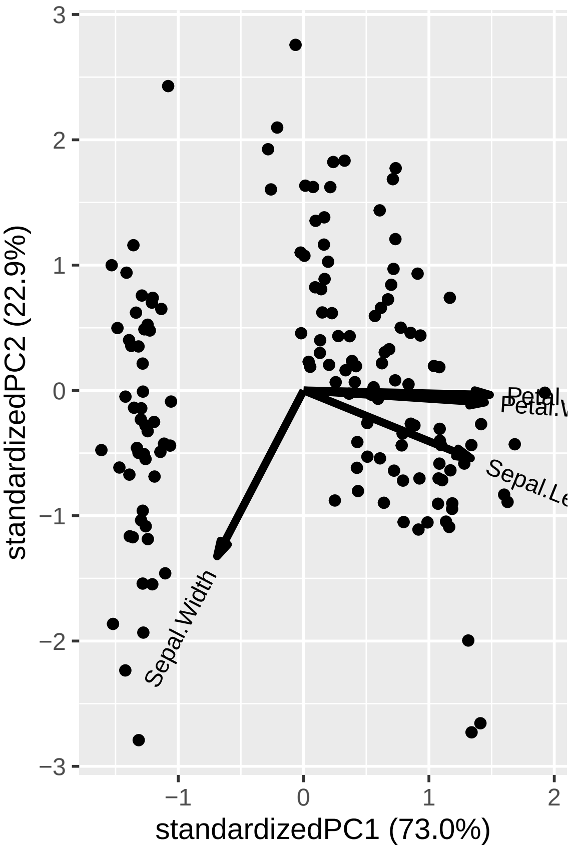
(a)

```
iris.pca <- prcomp(data,center=TRUE,scale. = TRUE)
summary(iris.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

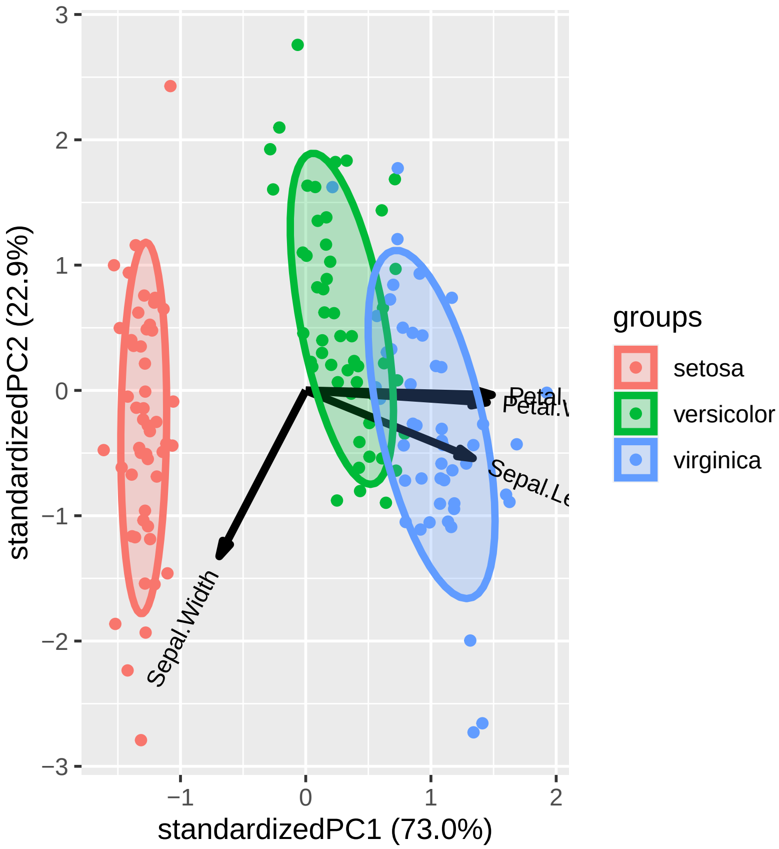
(b)

```
ggbiplot(iris.pca)
```



(c)

```
ggbiplot(iris.pca, ellipse = TRUE, groups = iris$Species)
```



(d)

```
iris.pca$rotation[,1] #PC1
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.5210659   -0.2693474    0.5804131    0.5648565
```

I recommend the PCA method as an efficient way here. It can reduce the dimension of the data, and the first two PC have a cumulative proportion of variance over 95%, which means this two PCs contains most of the information of the original variables. Additionally, the bi-plot created in part (c) would allows the biologists to see the groupings of species clearly.