Quiz 9

Srinivasa Phani Madhav Marupudi

4/16/2024

```r
library(tidyverse)
```

```
## -- 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 ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(neuralnet)
```

```
## Warning: package 'neuralnet' was built under R version 4.3.3
```

```
##
## Attaching package: 'neuralnet'
##
## The following object is masked from 'package:dplyr':
##
##     compute
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.3.3
```

```r
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.3.3
```

```
## 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
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(tidyverse)
library(glmnet)
```

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

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```r
library(ggplot2)
```

Q1

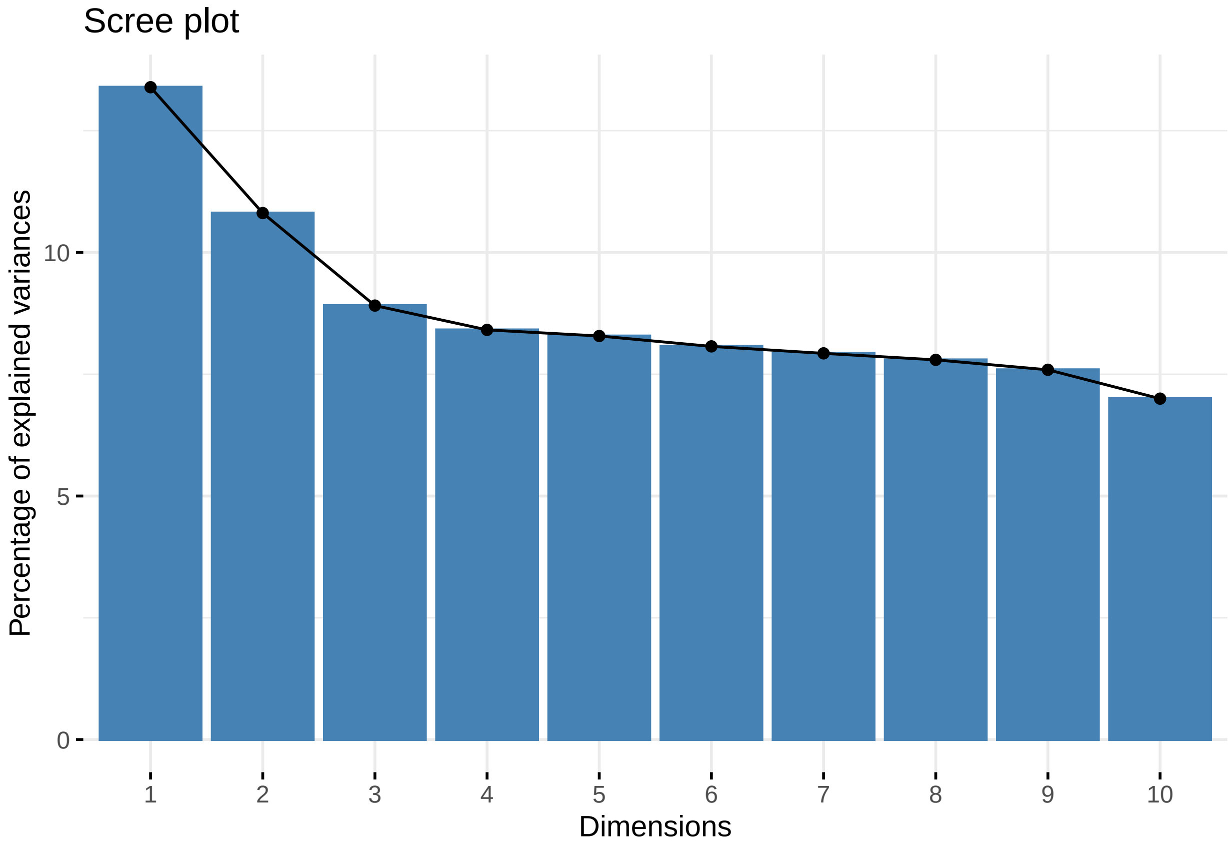
```
dat <- read.csv("C:/Users/MSP/Downloads/GreatUnknown(1).csv")
data <- na.omit(dat)
data = scale(data[,-13])
pc = princomp(data, cor = T)

library(factoextra)
```

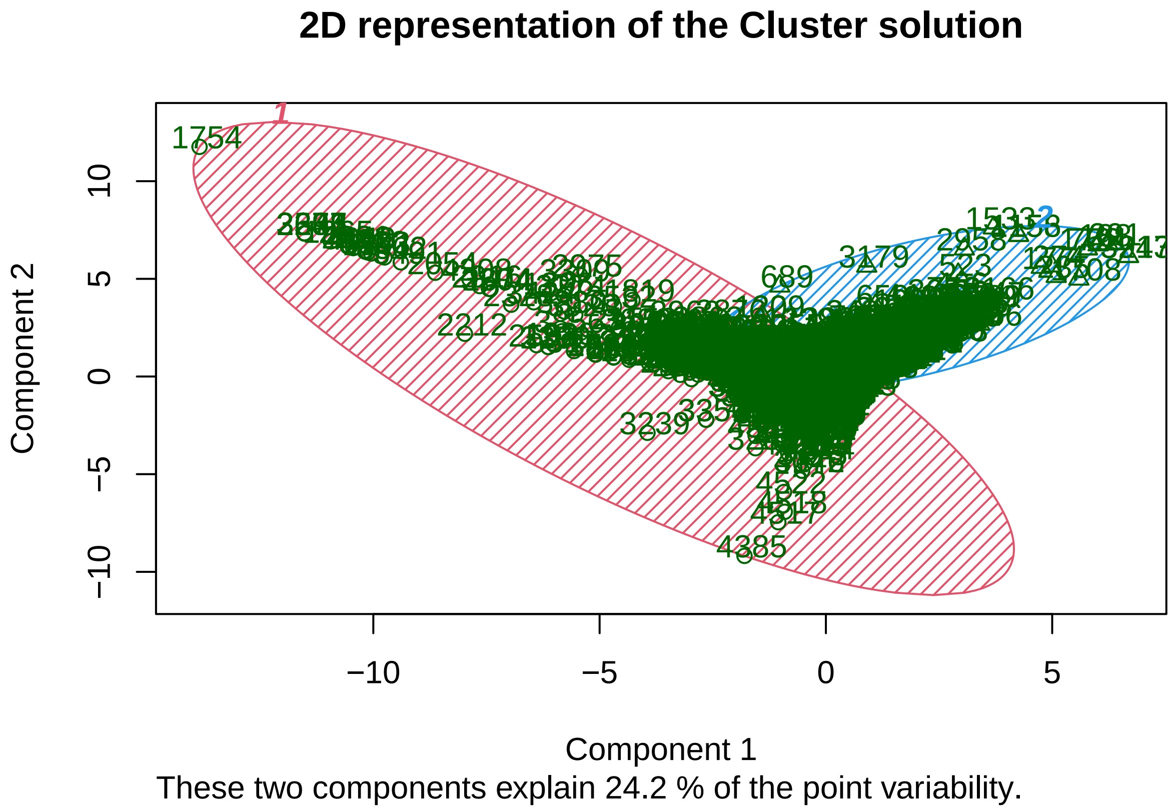
```
## Warning: package 'factoextra' was built under R version 4.3.3
```

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

```
fviz_eig(pc)
```



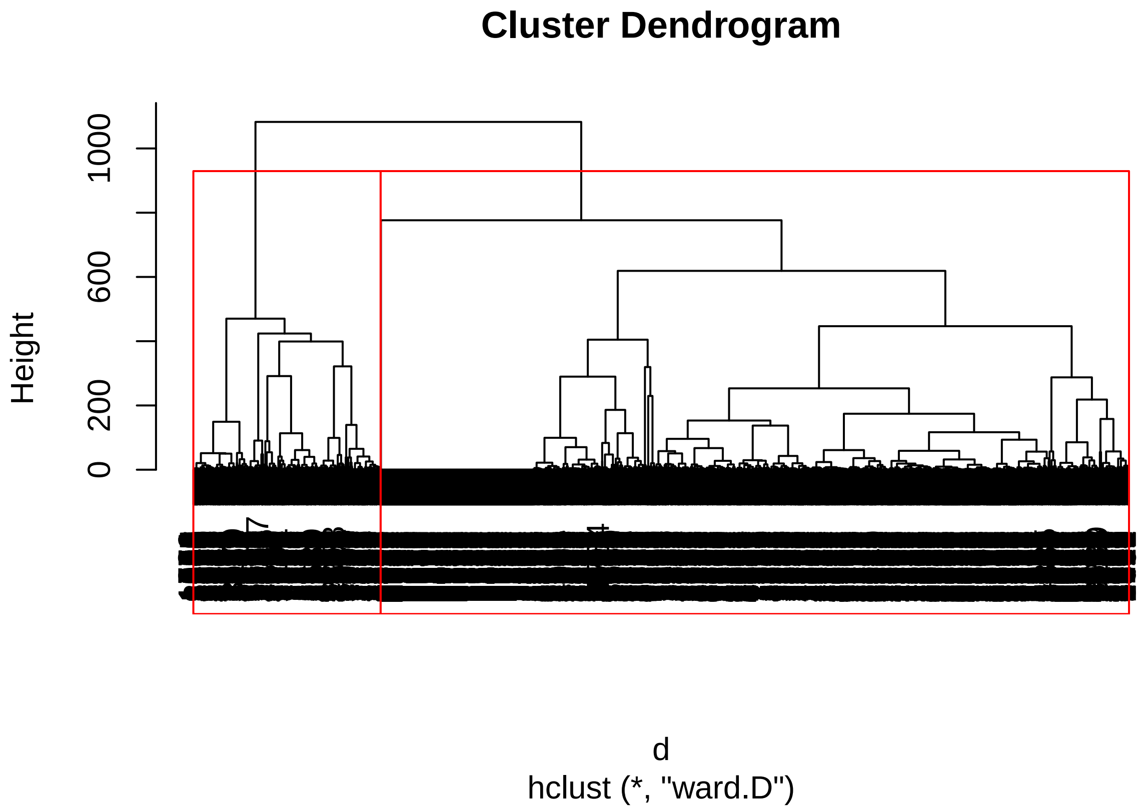
```
k.means.fit <- kmeans(data,2)
library(cluster)
clusplot(data, k.means.fit$cluster, main='2D representation of the Cluster solution',color=TRUE, shade='
```



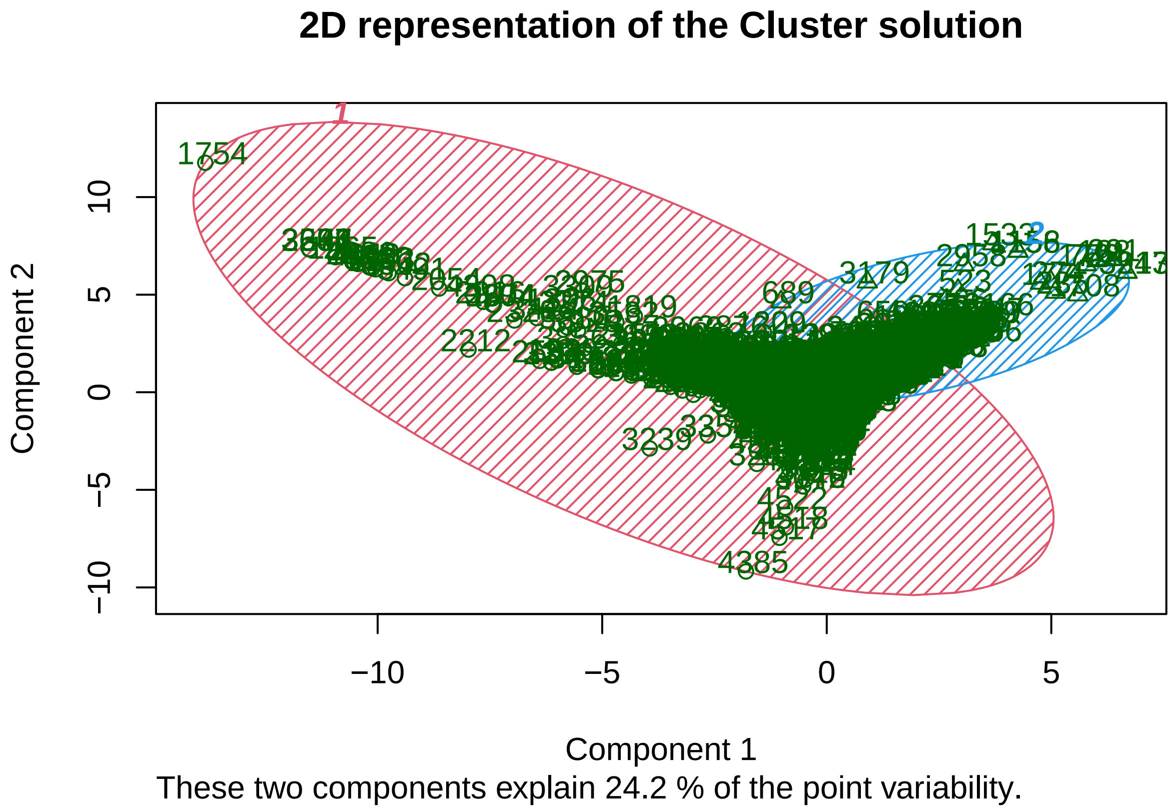
```r
table(k.means.fit$cluster,dat$y )
```

```
##
##         0    1
##   1 2643  900
##   2  145  913
```

```r
# H.ward
d <- dist(data, method = "euclidean")
H.fit <- hclust(d, method="ward.D")
plot(H.fit)
rect.hclust(H.fit, k=2, border="red")
```



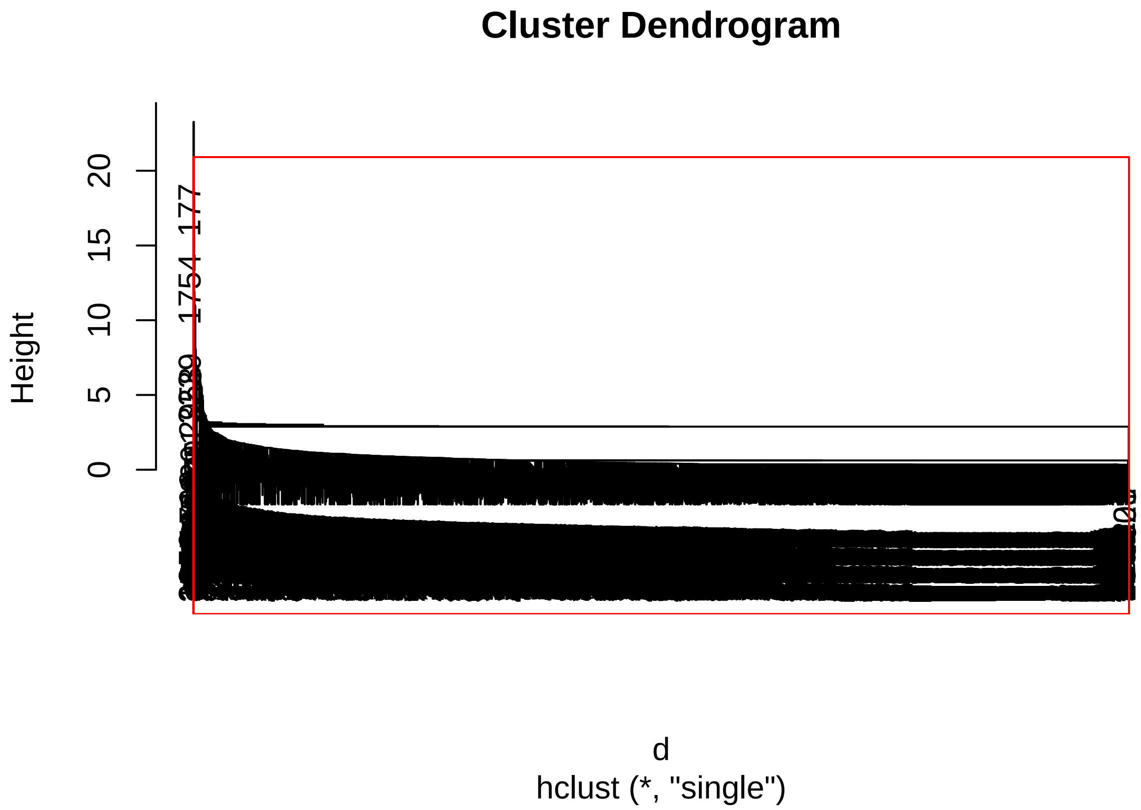
```
groups <- cutree(H.fit, k=2)
clusplot(data, groups, main='2D representation of the Cluster solution',color=TRUE, shade=TRUE,labels=2
```



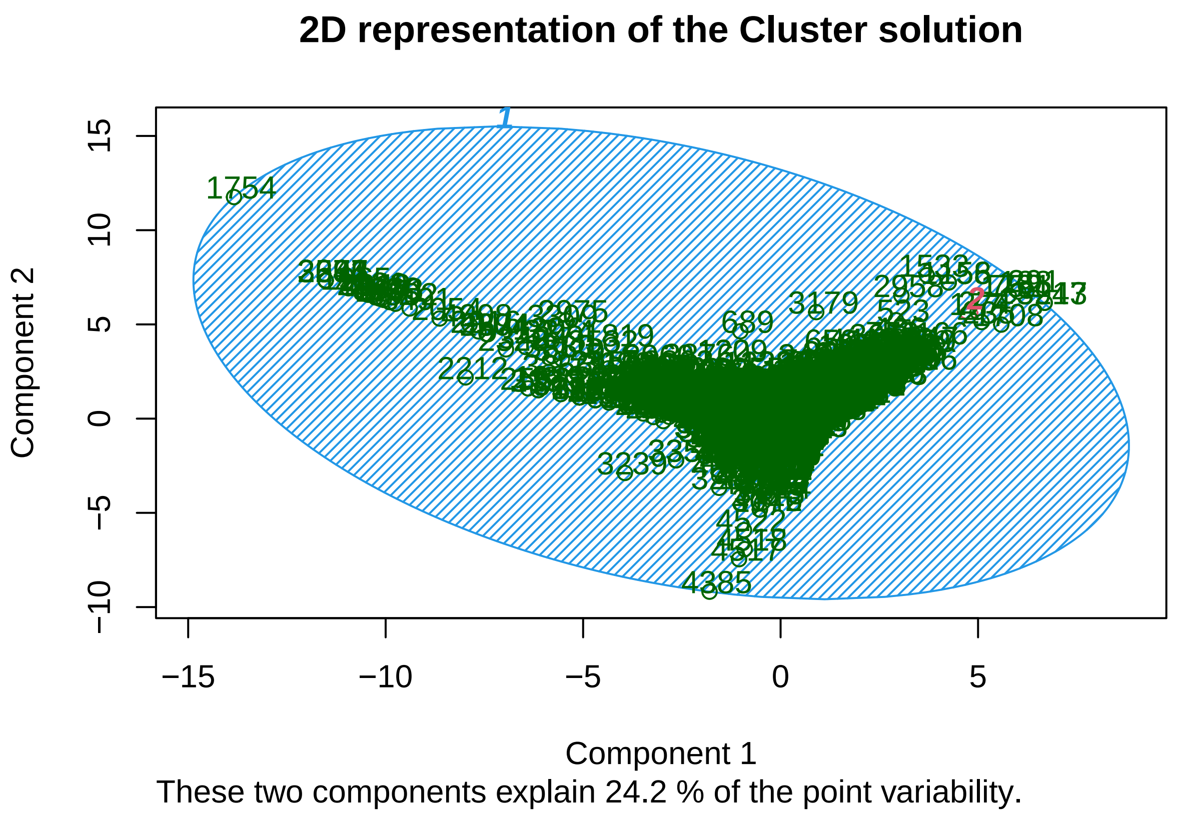
```r
clusters = factor(groups, levels = 1:2, labels = c("c1", "c2"))
table(dat[,13], clusters)
```

```
##    clusters
##       c1   c2
##   0 2621  167
##   1 1059  754
```

```r
# H.Single
H.fit <- hclust(d, method="single")
plot(H.fit)
rect.hclust(H.fit, k=2, border="red")
```



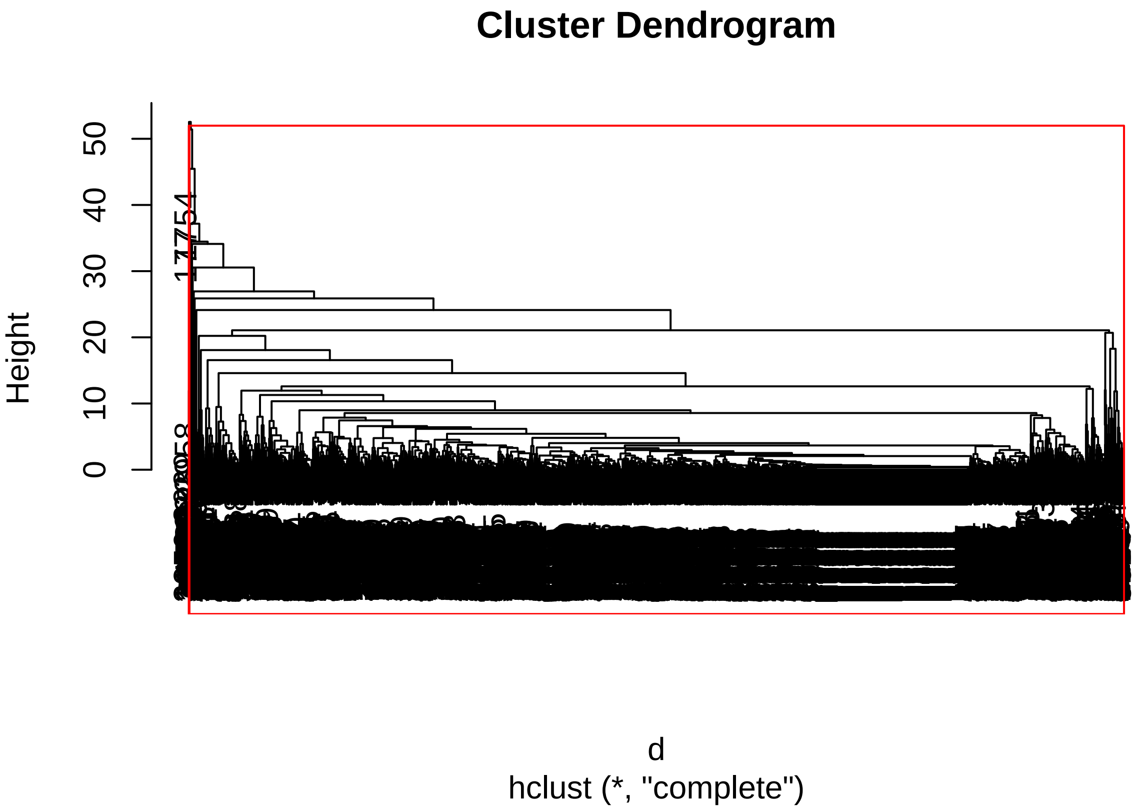
```
groups <- cutree(H.fit, k=2)
clusplot(data, groups, main='2D representation of the Cluster solution',color=TRUE, shade=TRUE,labels=2
```



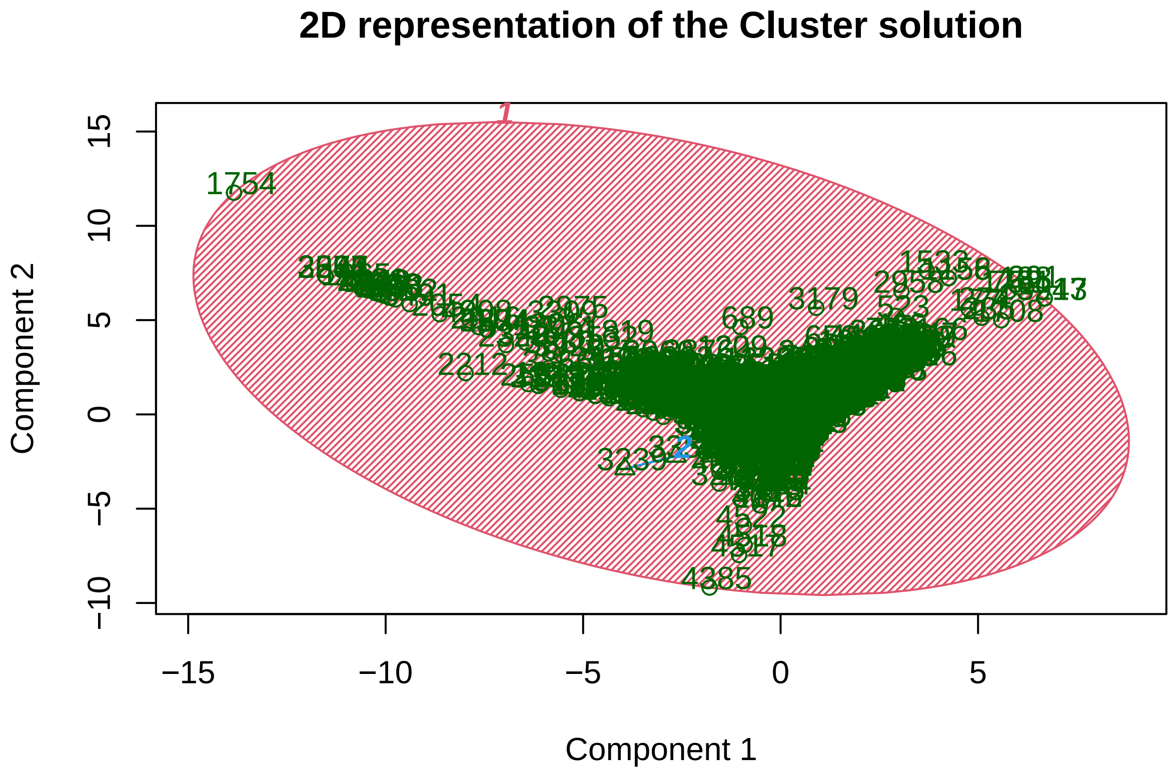
```
clusters = factor(groups, levels = 1:2, labels = c("c1", "c2"))
table(dat[,13], clusters)
```

```
##    clusters
##       c1   c2
##   0 2788    0
##   1 1812    1
```

```
# H.Complete
H.fit <- hclust(d, method="complete")
plot(H.fit)
rect.hclust(H.fit, k=2, border="red")
```



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

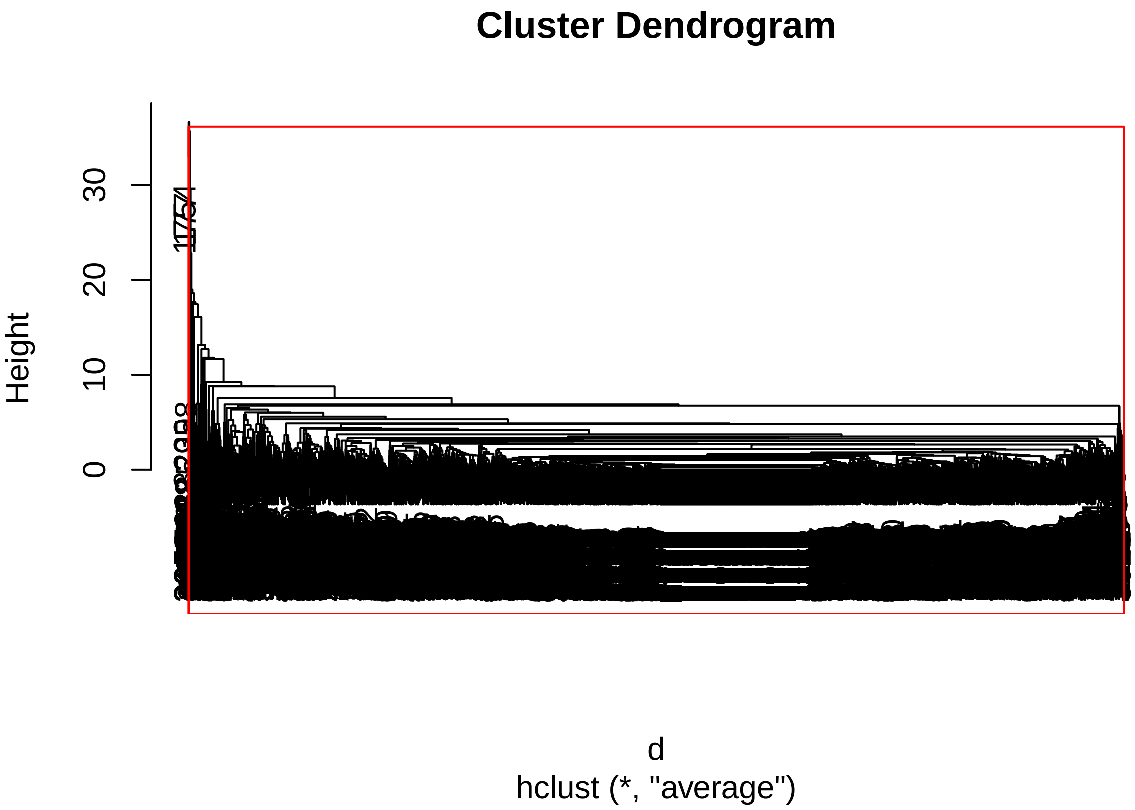


These two components explain 24.2 % of the point variability.

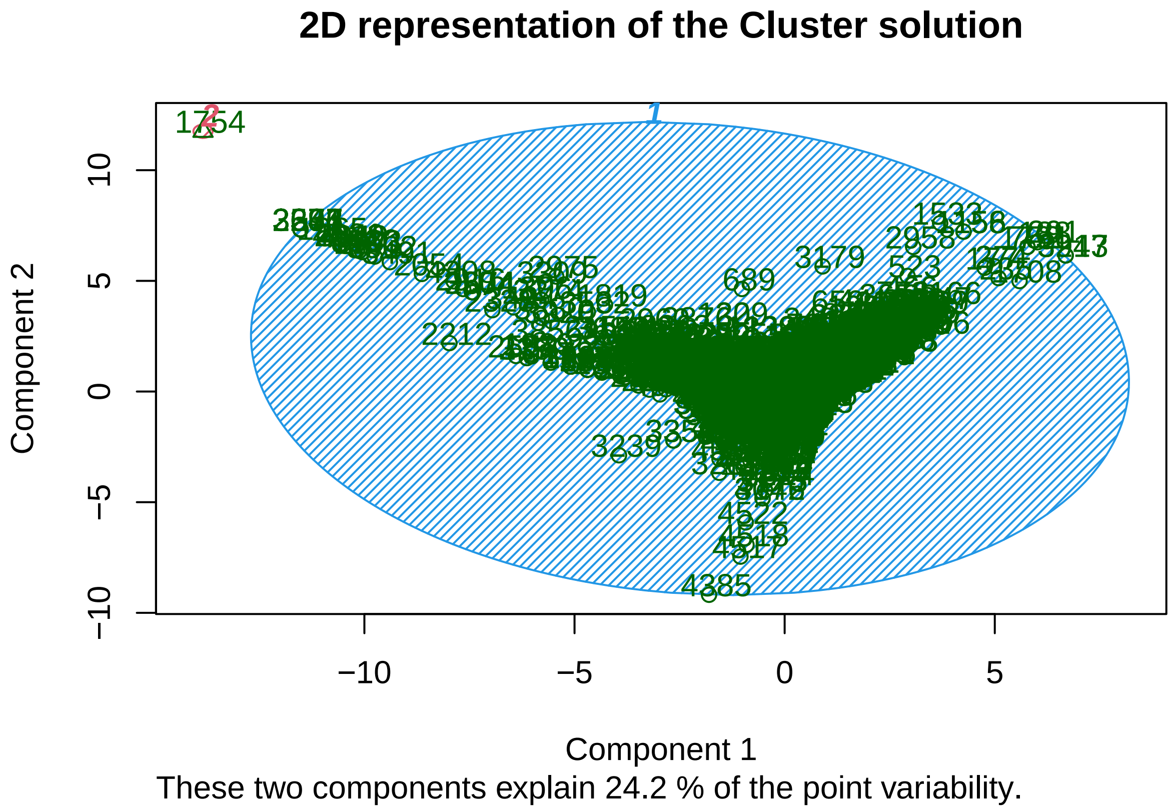
```
clusters = factor(groups, levels = 1:2, labels = c("c1", "c2"))
table(dat[,13], clusters)
```

```
##    clusters
##       c1   c2
##   0 2785    3
##   1 1813    0
```

```
# H.Average
H.fit <- hclust(d, method="average")
plot(H.fit)
rect.hclust(H.fit, k=2, border="red")
```



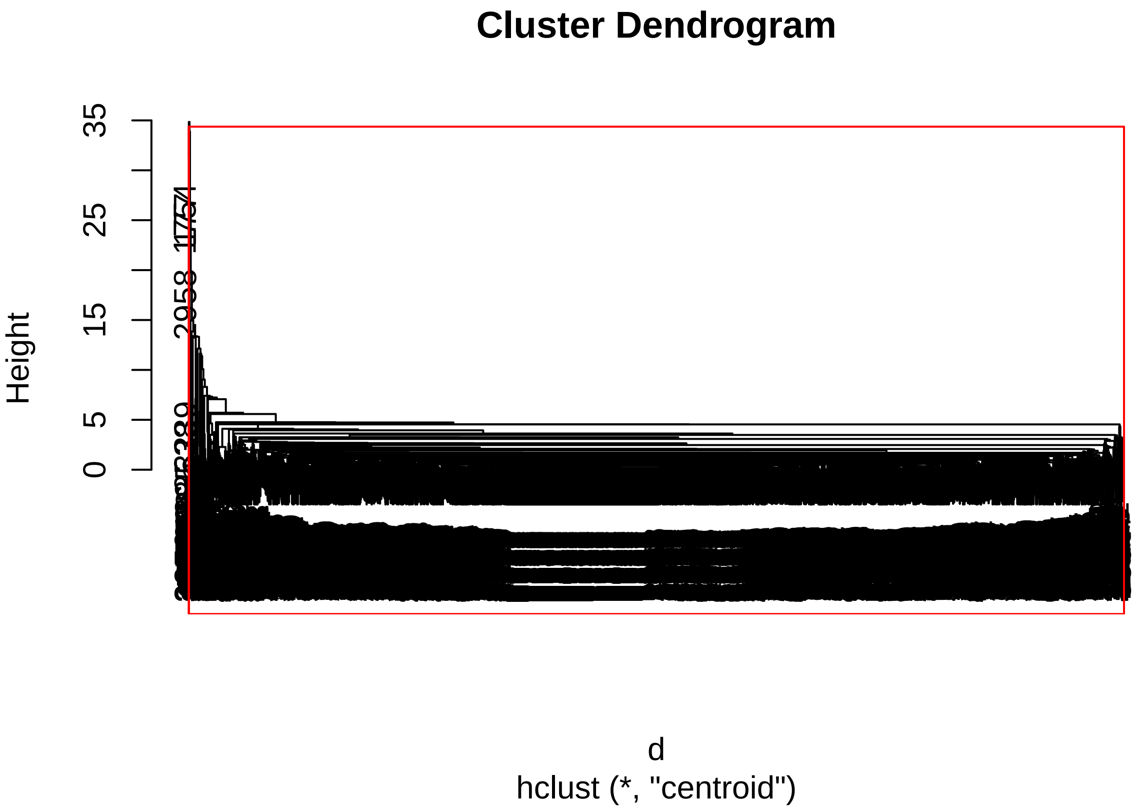
```
groups <- cutree(H.fit, k=2)
clusplot(data, groups, main='2D representation of the Cluster solution',color=TRUE, shade=TRUE,labels=2
```



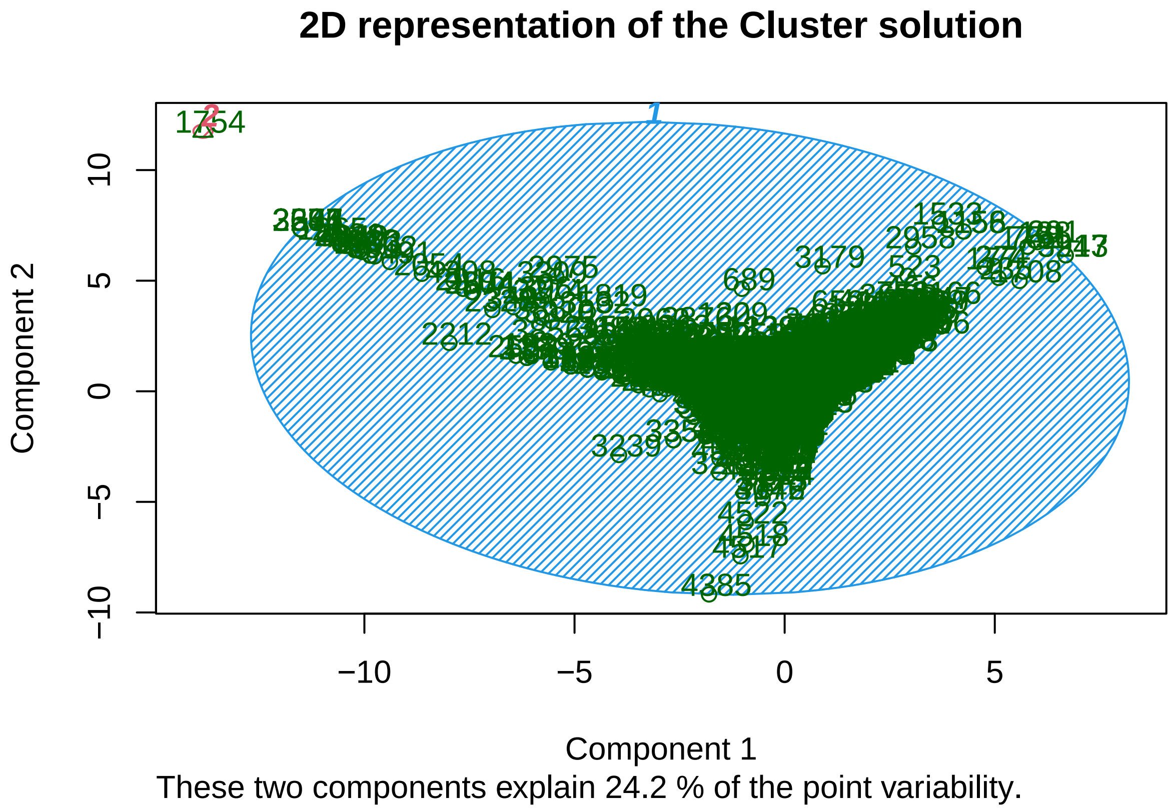
```r
clusters = factor(groups, levels = 1:2, labels = c("c1", "c2"))
table(dat[,13], clusters)
```

```
##    clusters
##       c1   c2
##   0 2788    0
##   1 1812    1
```

```r
# H.Centroid
H.fit <- hclust(d, method="centroid")
plot(H.fit)
rect.hclust(H.fit, k=2, border="red")
```



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



```
clusters = factor(groups, levels = 1:2, labels = c("c1", "c2"))
table(dat[,13], clusters)
```

```
##    clusters
##       c1   c2
##   0 2788    0
##   1 1812    1
```

```
#(5) Wards is better than k means by a sizeable margin.
#(6)
# In this case Single = Average = Centroid > Wards method > complete > k-means in terms of accuracy and
```

Q2

```
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 4.3.3
```

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

```
## Warning: package 'usethis' was built under R version 4.3.3
```

```
library(ggbiplot)
```

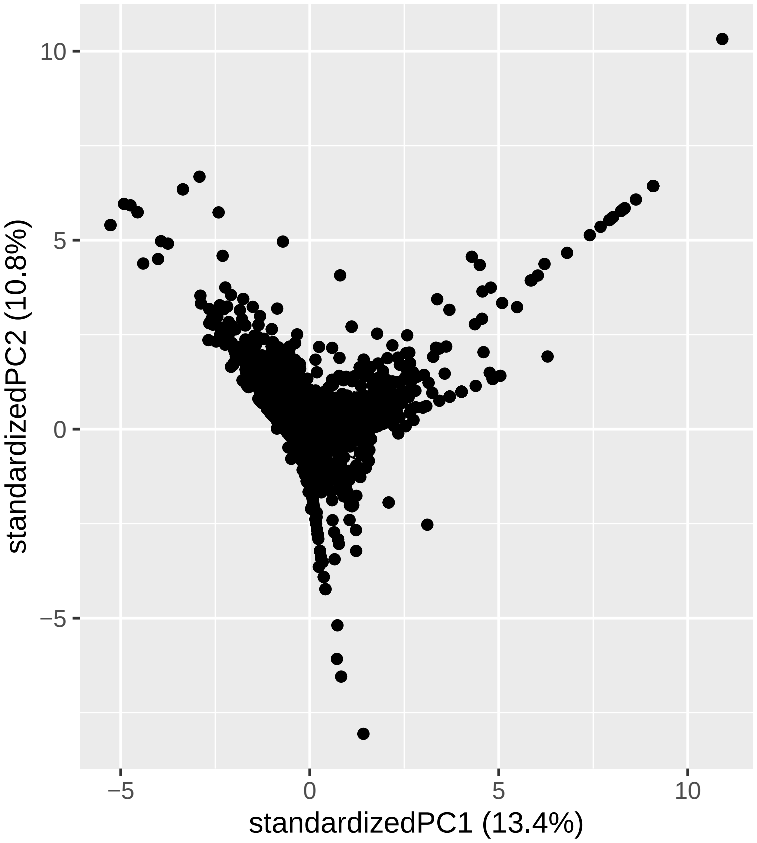
```
## Warning: package 'ggbiplot' was built under R version 4.3.3
```

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

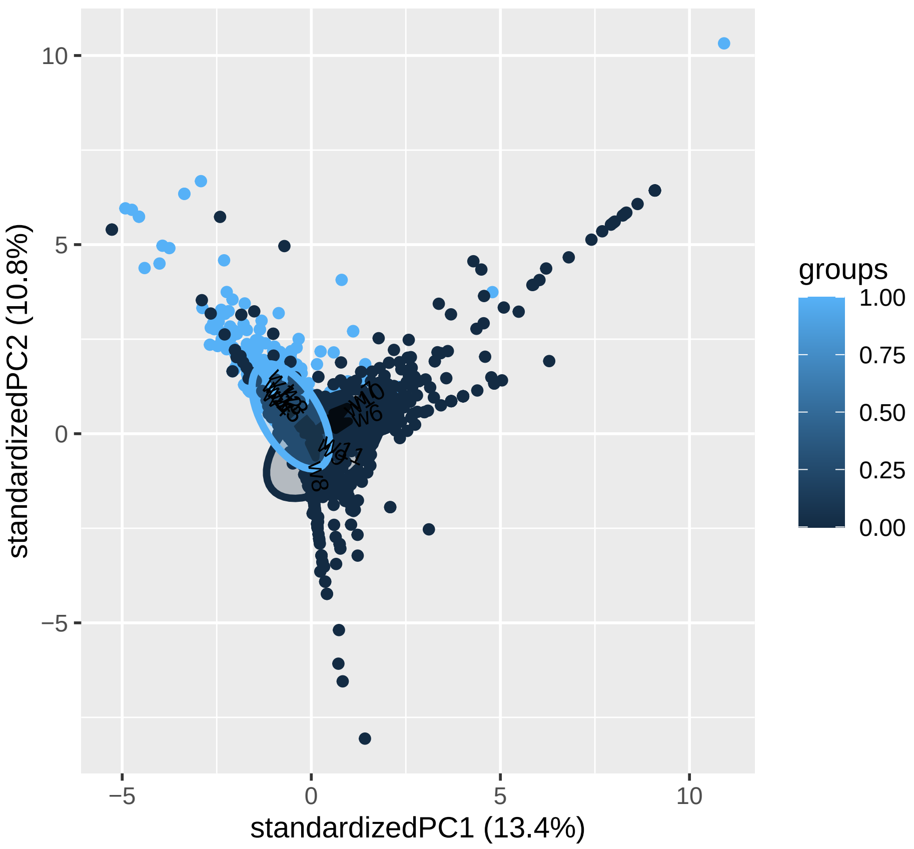
```
#(a)
dat.pca = prcomp(data, center = TRUE,scale. = TRUE)
summary(dat.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.2677 1.1389 1.0340 1.00462 0.99711 0.98418 0.97543
## Proportion of Variance 0.1339 0.1081 0.0891 0.08411 0.08285 0.08072 0.07929
## Cumulative Proportion  0.1339 0.2420 0.3311 0.41522 0.49807 0.57879 0.65807
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.96718 0.95444 0.91643 0.89253 0.78756
## Proportion of Variance 0.07795 0.07591 0.06999 0.06638 0.05169
## Cumulative Proportion  0.73603 0.81194 0.88193 0.94831 1.00000
```

```
#(b)
ggbiplot(dat.pca)
```



```
#(c)
ggbiplot(dat.pca, ellipse=TRUE, groups=dat$y)
```



```
#(d)
dat.pca$rotation[,1]
```

```
##          w1          w2          w3          w4          w5          w6
## -0.26312837 -0.30497476 -0.10446981 -0.27139508 -0.13025875  0.52197111
##          w7          w8          w9         w10         w11         w12
##  0.49664611  0.05810733  0.08953016  0.40916977  0.10868236 -0.16273359
```

```
# We can see that w6,w7,w8,w9,w10,w11 have positive correlation , while the rest have a negative correl
#We can deduce that considering upto PC10 should be sufficient for most cases as they have a cumilative
```