zalshaye_4

```
# load libraries
library(tidyverse) # group of packages to wrangle and visualize data

## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.3

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(cluster) # cluster analysis
library(factoextra) # visualize clusters and principal components

## Warning: package 'factoextra' was built under R version 4.0.3

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

library(dendextend) # visualize dendrograms

## Warning: package 'dendextend' was built under R version 4.0.3

##
## --------------------
## Welcome to dendextend version 1.14.0
## 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:
```

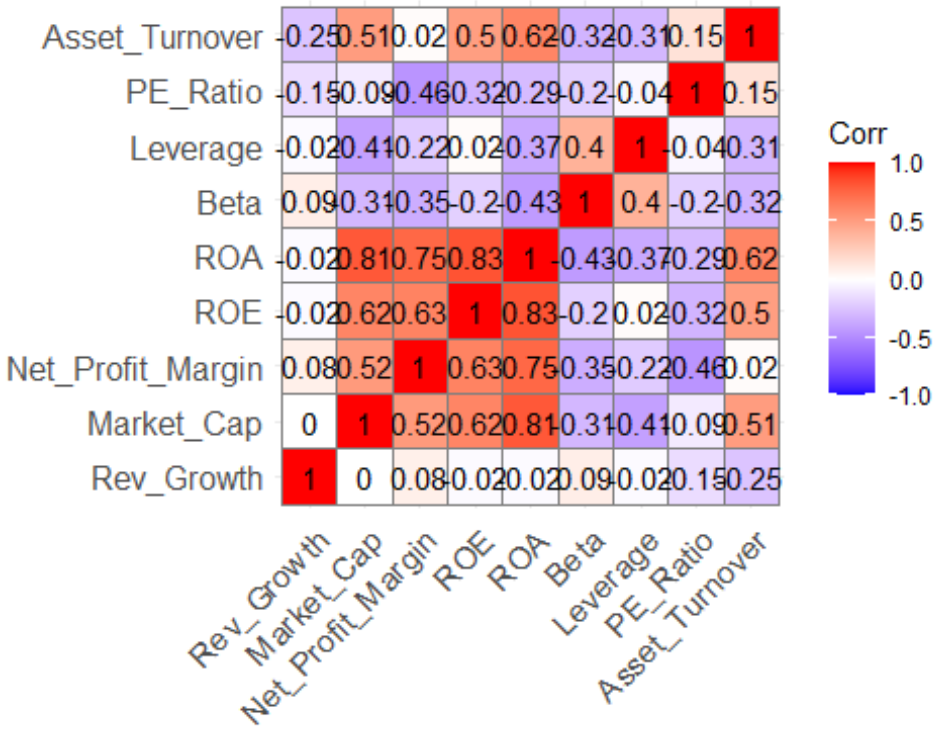
```
suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'

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

library(here) # create a file directory

## Warning: package 'here' was built under R version 4.0.5

## here() starts at C:/Users/Z/Desktop

library(ggrepel) # repel overlapping text labels

## Warning: package 'ggrepel' was built under R version 4.0.3

library(clustree) # visualize clusters

## Warning: package 'clustree' was built under R version 4.0.5

## Loading required package: ggraph

## Warning: package 'ggraph' was built under R version 4.0.5

library(FactoMineR) # explore multivariate data

## Warning: package 'FactoMineR' was built under R version 4.0.3

library(ggcorrplot) # visualize correlations

## Warning: package 'ggcorrplot' was built under R version 4.0.5

library(clValid) # compute cluster metrics

## Warning: package 'clValid' was built under R version 4.0.5

library(broom) # tidy algorithm outputs

## Warning: package 'broom' was built under R version 4.0.3

library(umap) # dimension reduction

## Warning: package 'umap' was built under R version 4.0.5

library(tidyquant) # in this case theme and color for clusters visualization

## Warning: package 'tidyquant' was built under R version 4.0.5

## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: PerformanceAnalytics

## Warning: package 'PerformanceAnalytics' was built under R version 4.0.5

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.0.5

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## Loading required package: quantmod

## Warning: package 'quantmod' was built under R version 4.0.5

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## == Need to Learn tidyquant?
==================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance
Analysis & Portfolio Optimization with tidyquant!
```

```
## </> Learn more at: https://university.business-science.io/p/learning-labs-
pro </>

# load the file
Pharmaceuticals <- read_csv(here::here("C:/Users/Z/Desktop/Fall
2021/ML/Assignment 4/Pharmaceuticals.csv"))

##
## -- Column specification --------------------------------------------------
------
## cols(
##   Symbol = col_character(),
##   Name = col_character(),
##   Market_Cap = col_double(),
##   Beta = col_double(),
##   PE_Ratio = col_double(),
##   ROE = col_double(),
##   ROA = col_double(),
##   Asset_Turnover = col_double(),
##   Leverage = col_double(),
##   Rev_Growth = col_double(),
##   Net_Profit_Margin = col_double(),
##   Median_Recommendation = col_character(),
##   Location = col_character(),
##   Exchange = col_character()
## )

# explore the data
glimpse(Pharmaceuticals)

## Rows: 21
## Columns: 14
## $ Symbol                <chr> "ABT", "AGN", "AHM", "AZN", "AVE", "BAY",
"BMY",~
## $ Name                  <chr> "Abbott Laboratories", "Allergan, Inc.",
"Amersh~
## $ Market_Cap            <dbl> 68.44, 7.58, 6.30, 67.63, 47.16, 16.90,
51.33, 0~
## $ Beta                  <dbl> 0.32, 0.41, 0.46, 0.52, 0.32, 1.11, 0.50,
0.85, ~
## $ PE_Ratio              <dbl> 24.7, 82.5, 20.7, 21.5, 20.1, 27.9, 13.9,
26.0, ~
## $ ROE                   <dbl> 26.4, 12.9, 14.9, 27.4, 21.8, 3.9, 34.8,
24.1, 1~
## $ ROA                   <dbl> 11.8, 5.5, 7.8, 15.4, 7.5, 1.4, 15.1, 4.3,
5.1, ~
## $ Asset_Turnover        <dbl> 0.7, 0.9, 0.9, 0.9, 0.6, 0.6, 0.9, 0.6, 0.3,
0.6~
## $ Leverage              <dbl> 0.42, 0.60, 0.27, 0.00, 0.34, 0.00, 0.57,
3.51, ~
## $ Rev_Growth            <dbl> 7.54, 9.16, 7.05, 15.00, 26.81, -3.17, 2.70,
```

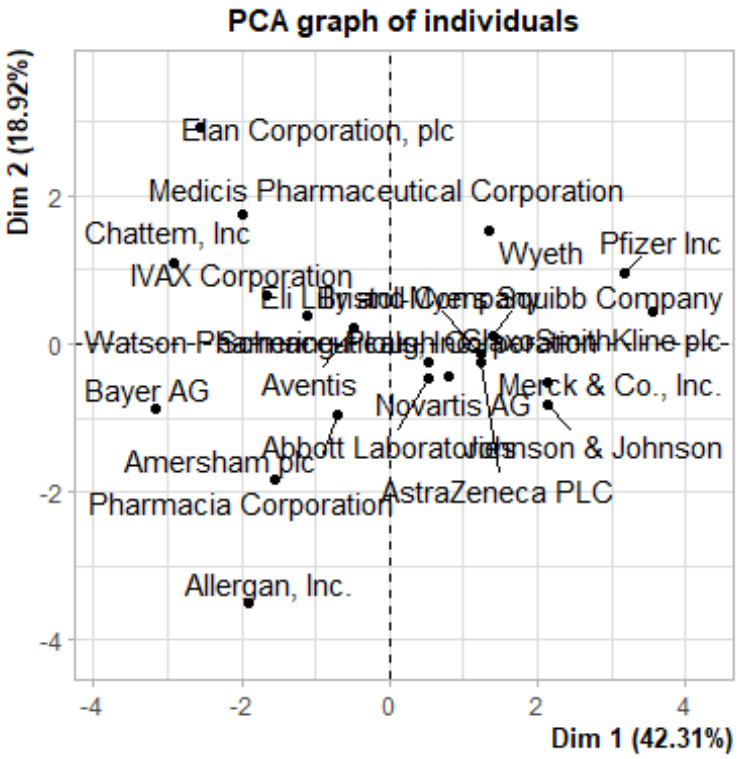
```
6.3~
## $ Net_Profit_Margin     <dbl> 16.1, 5.5, 11.2, 18.0, 12.9, 2.6, 20.6, 7.5,
13.~
## $ Median_Recommendation <chr> "Moderate Buy", "Moderate Buy", "Strong
Buy", "M~
## $ Location              <chr> "US", "CANADA", "UK", "UK", "FRANCE",
"GERMANY",~
## $ Exchange              <chr> "NYSE", "NYSE", "NYSE", "NYSE", "NYSE",
"NYSE", ~
```

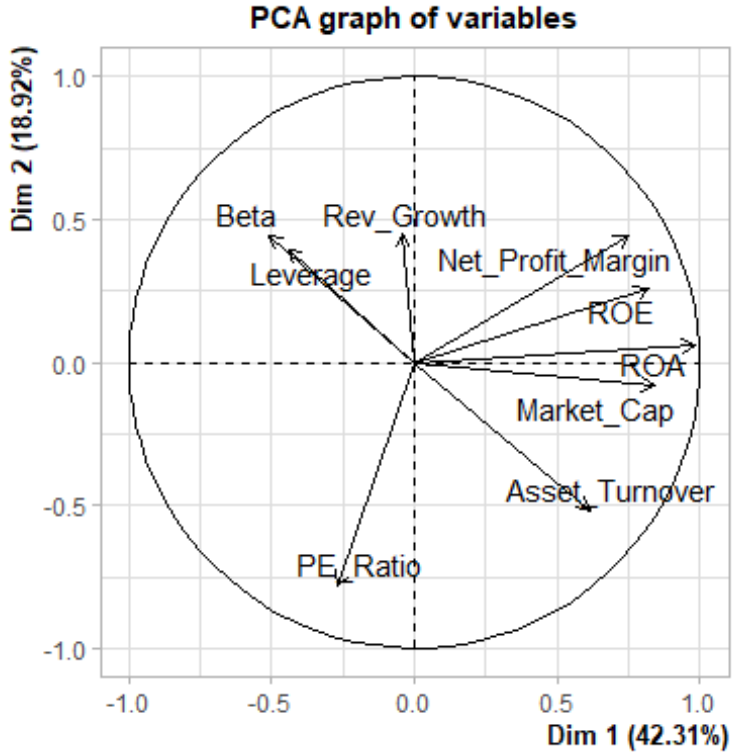
```
# create correlation matrix
Pharmaceuticals_cor <- Pharmaceuticals %>%
  select_if(is.numeric) %>%
  cor()

# visualize correlations
ggcorrplot(Pharmaceuticals_cor,
           outline.color = "grey50",
           lab = TRUE,
           hc.order = TRUE,
           type = "full")
```




```
# use a data frame only with numeric values and scale the variables because
they were measured in different scales
Pharmaceuticals_tbl <- na.omit(Pharmaceuticals) %>%
dplyr::select(-c(1, 12, 13, 14)) %>%
column_to_rownames(var = "Name") %>%
```

```
scale(.) %>% # standardize the values
as.data.frame() # convert to data frame

## use PCA to check how many dimensions we have
# PCA of our dataframe
new_pca <- PCA(Pharmaceuticals_tbl)
```

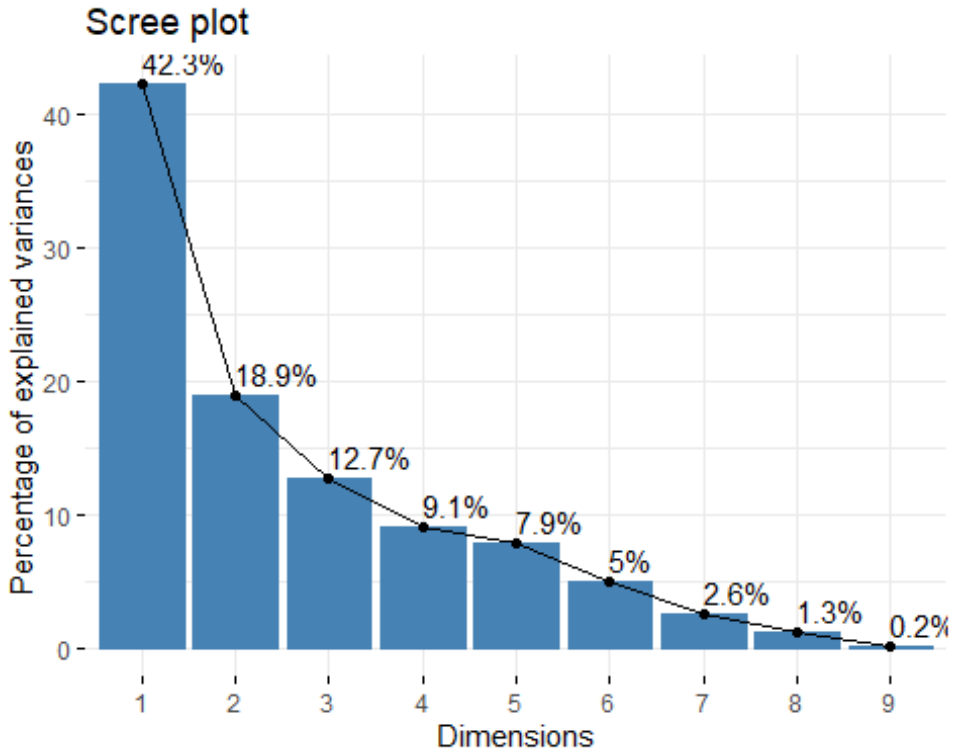




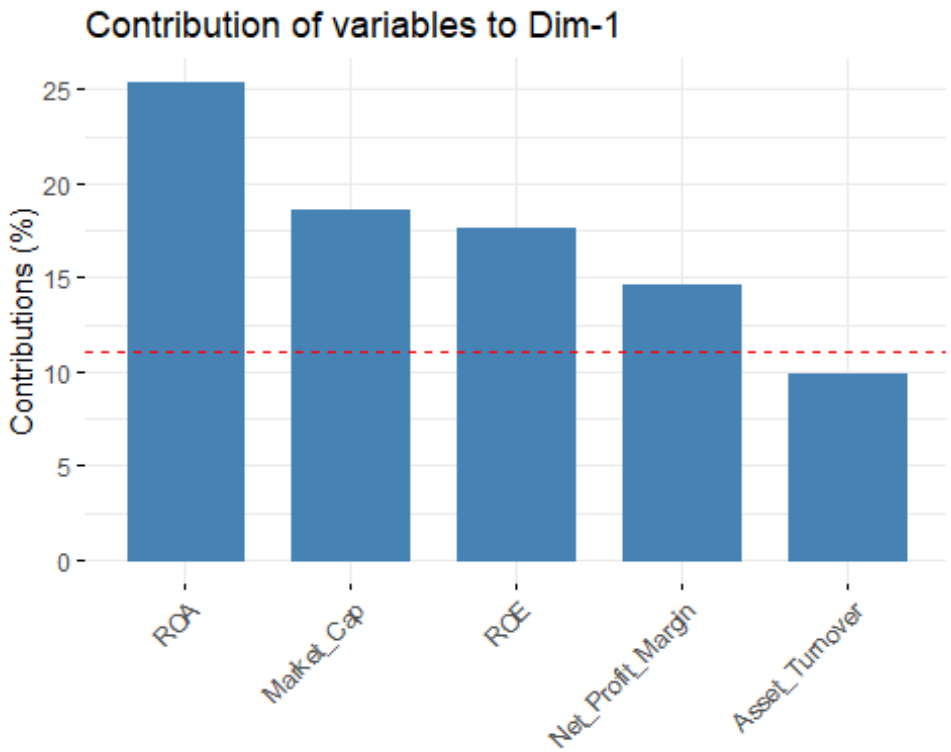

```
# check eigenvalues and percentage of variance
new_pca$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  3.8080296             42.3114401                           42.31144
## comp 2  1.7028349             18.9203881                           61.23183
## comp 3  1.1435807             12.7064523                           73.93828
## comp 4  0.8157384              9.0637604                           83.00204
## comp 5  0.7071235              7.8569272                           90.85897
## comp 6  0.4538979              5.0433098                           95.90228
## comp 7  0.2337408              2.5971197                           98.49940
## comp 8  0.1154565              1.2828502                           99.78225
## comp 9  0.0195977              0.2177522                          100.00000
```

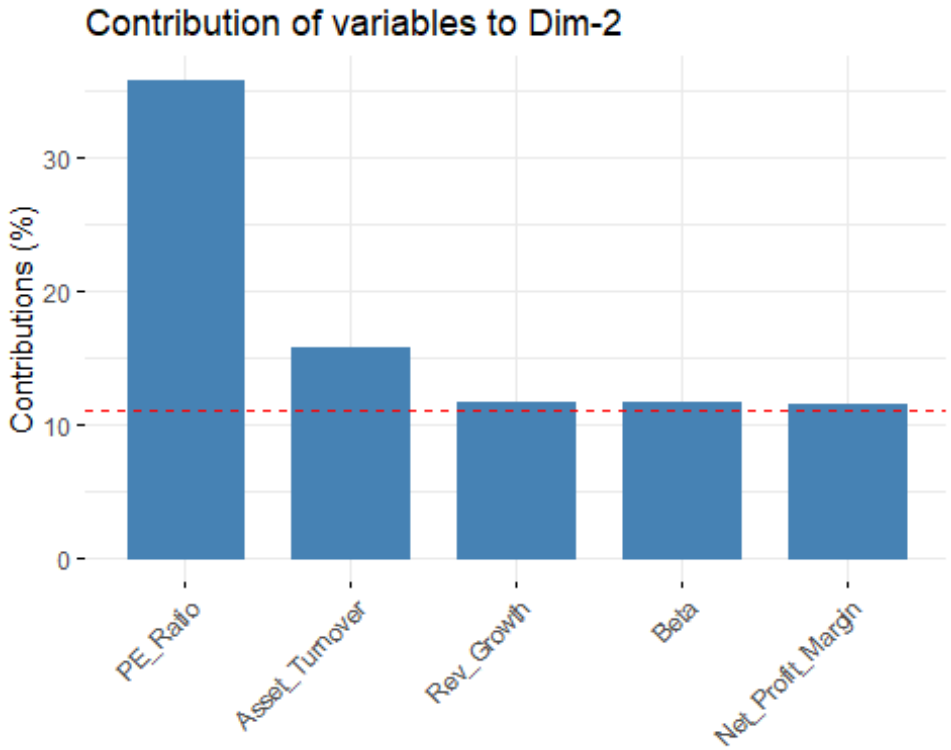
```r
# visualization of how much variance each dimension explains
fviz_screeplot(new_pca, addlabels = TRUE)
```



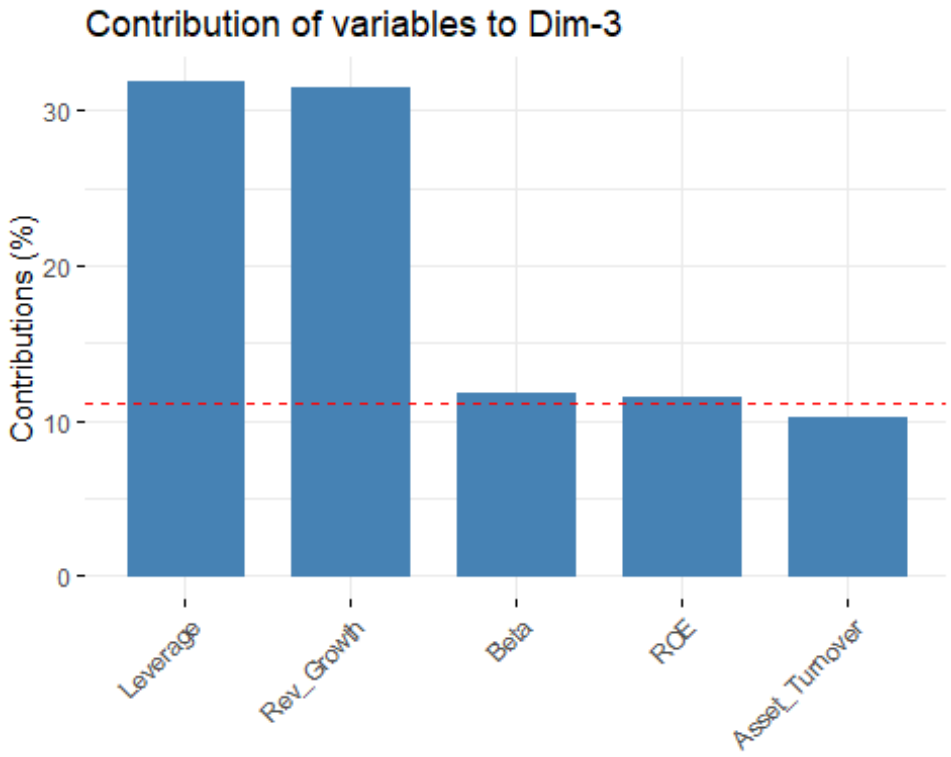

```r
# get each variable PCA results
var <- get_pca_var(new_pca)

# each variable contribution to PC1 - top 5
fviz_contrib(new_pca, choice = "var", axes = 1, top = 5)
```



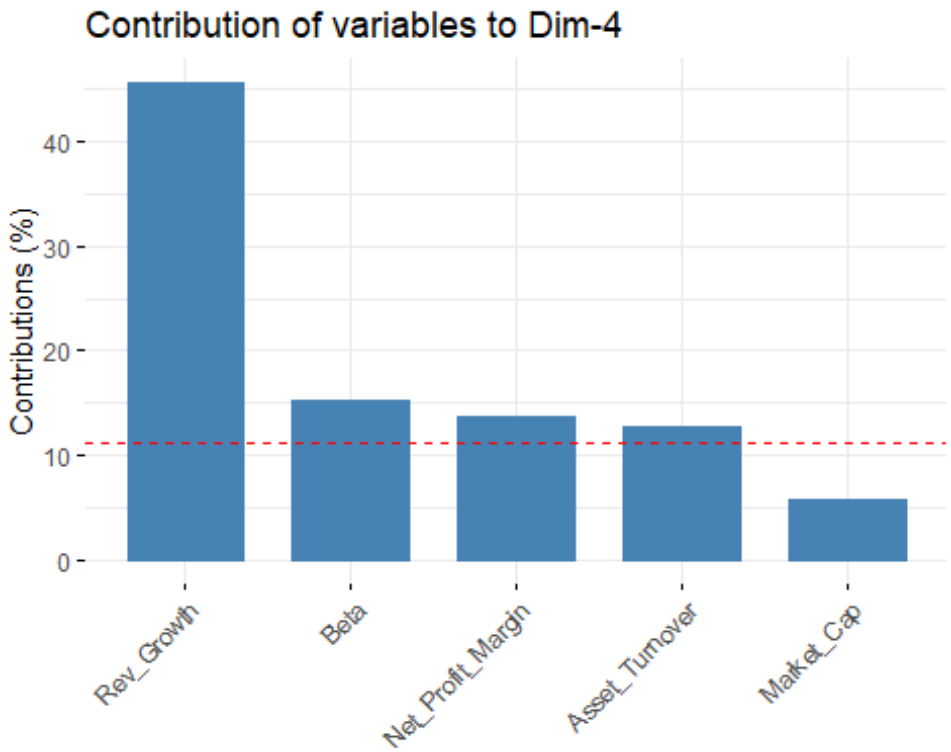

```r
# each variable contribution to PC2 - top 5
fviz_contrib(new_pca, choice = "var", axes = 2, top = 5)
```



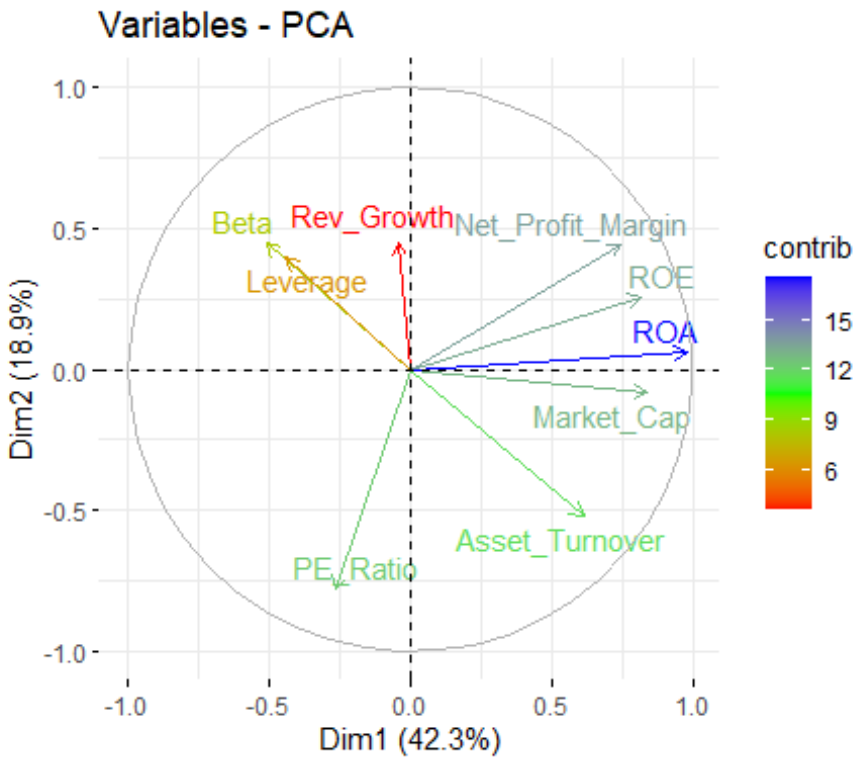
```r
# each variable contribution to PC3 - top 5
fviz_contrib(new_pca, choice = "var", axes = 3, top = 5)
```



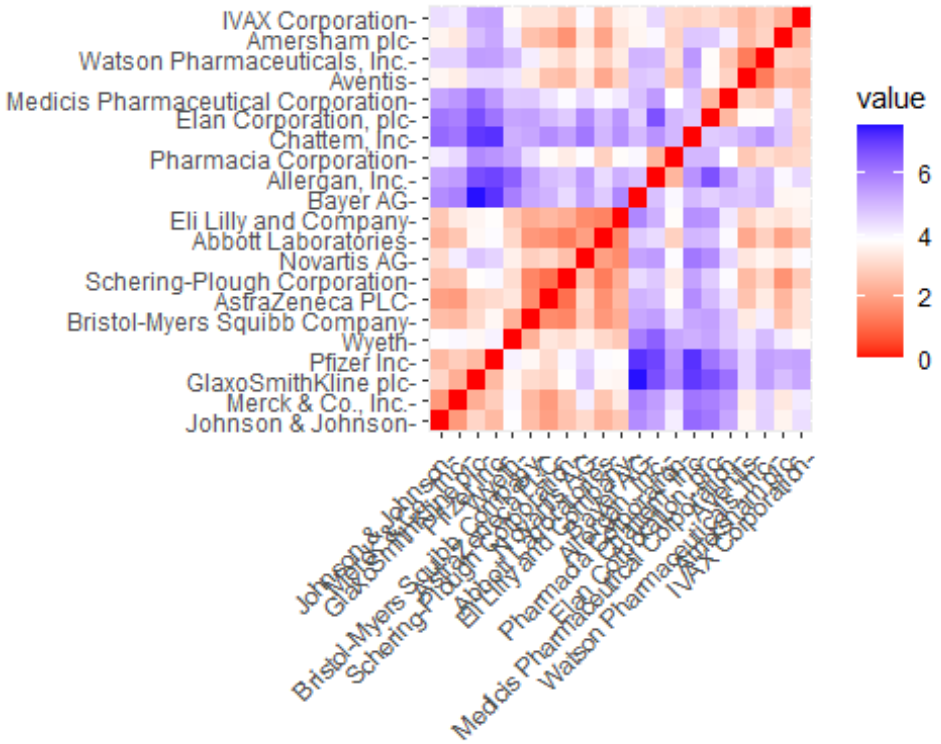

```r
# each variable contribution to PC - top 5
fviz_contrib(new_pca, choice = "var", axes = 4, top = 5)
```



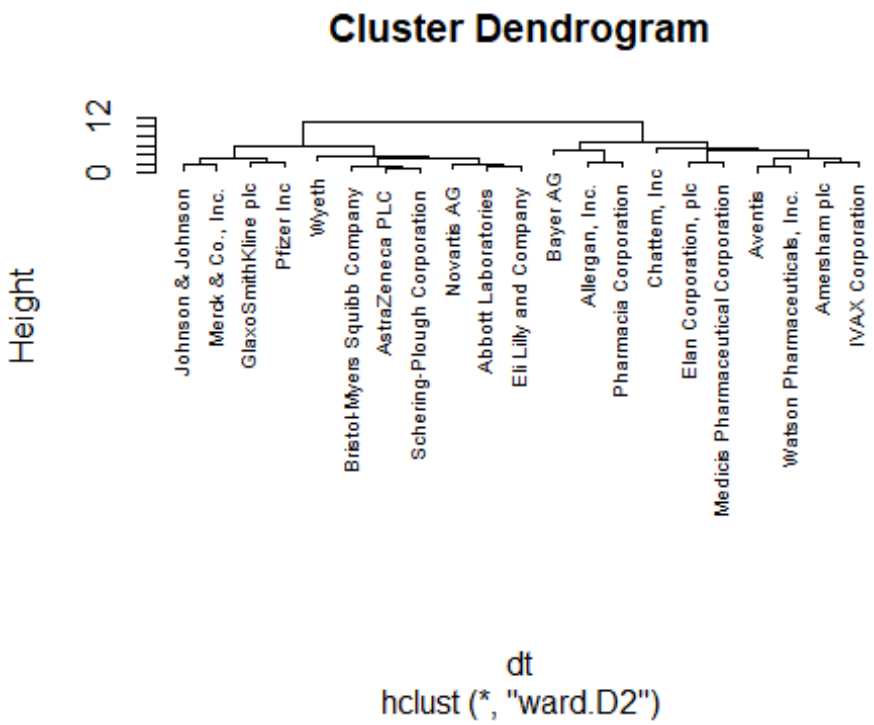

```
# visualization of the first two components and the contributions of each
variable
fviz_pca_var(new_pca, col.var="contrib",
gradient.cols = c("red", "green", "blue"),
repel = TRUE
) +
labs( title = "Variables - PCA")
```



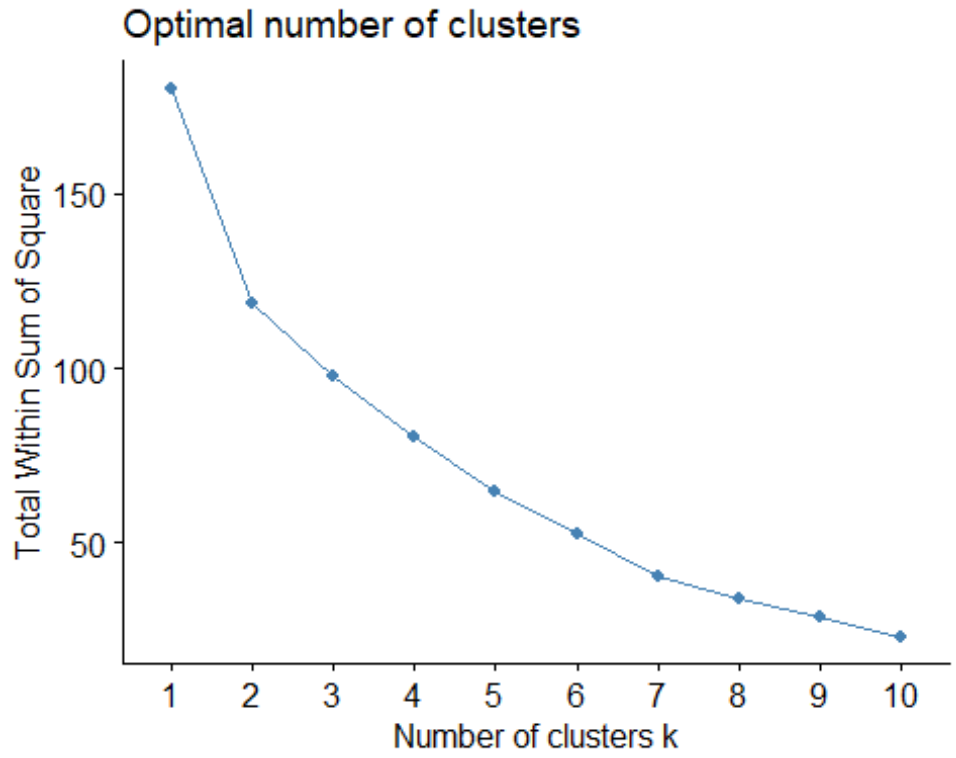

```
## Hierarchical Cluster Analysis

# compute distance measure
dt <- dist(Pharmaceuticals_tbl, method = "euclidean")


# visualize distance
fviz_dist(dt, gradient = list(low = "red", mid = "white", high = "blue"))
```



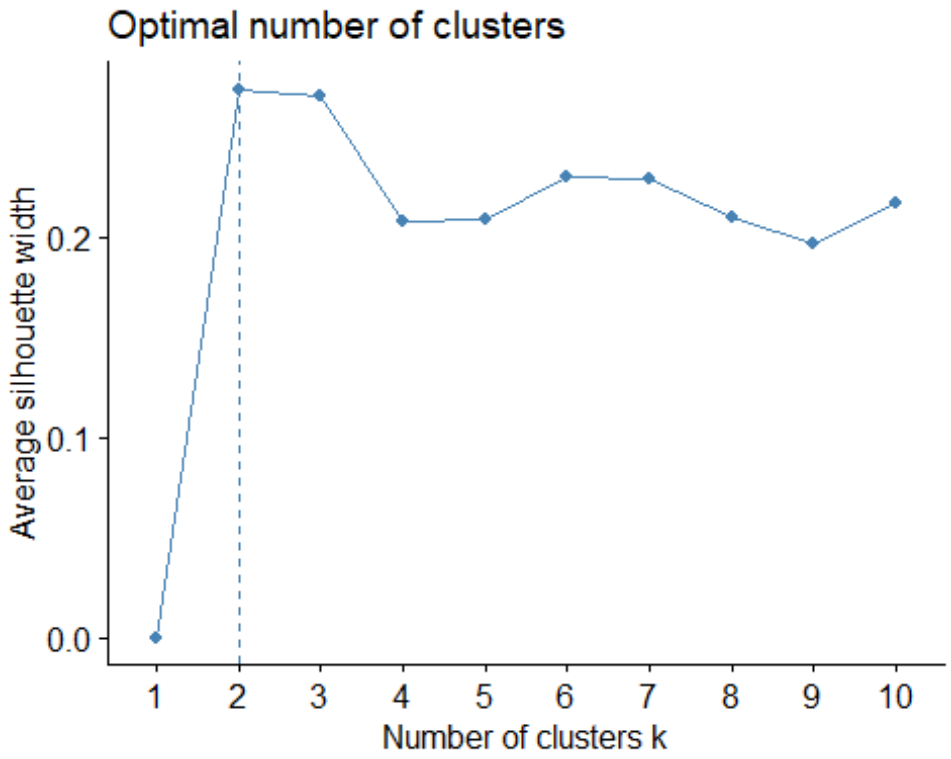

```
m <- c("average", "single", "complete", "ward")
names(m) <- c("average", "single", "complete", "ward")


#function to check the best (means higher value) linkage method
ac <- function(x) {
  agnes(dt, method = x)$ac
}

map_dbl(m, ac)

##   average    single  complete      ward
## 0.5600652 0.4600348 0.6990833 0.7943164

# hierarchical clustering
set.seed(88)
hclust_1 <- hclust(dt, method = "ward.D2") # ward.D2 corresponds to the ward
method in the hclust function

# plot hierarchical clustering
plot(hclust_1, cex = 0.6)
```



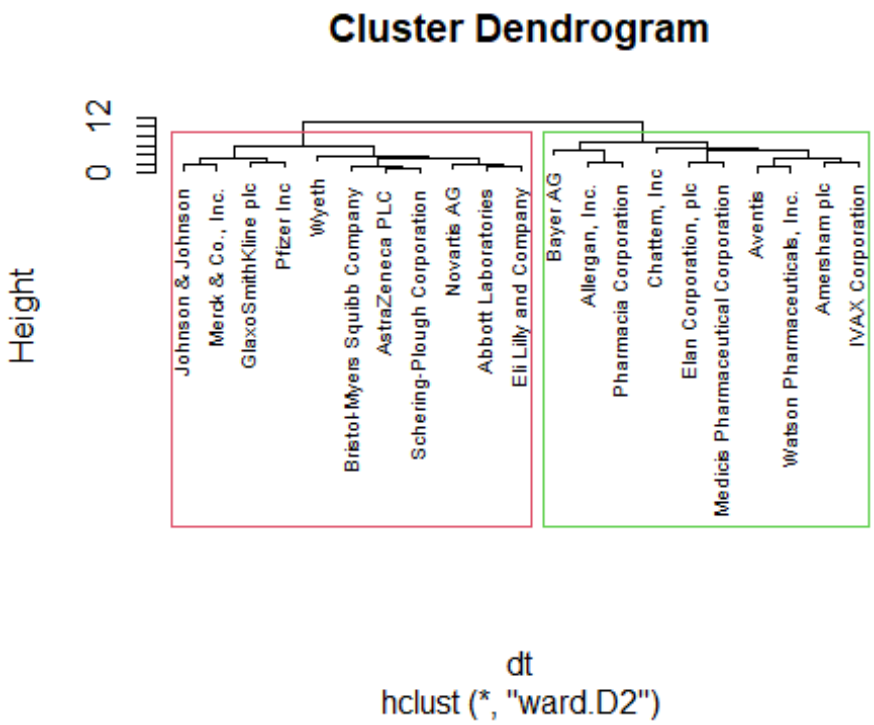

```
# elbow method
fviz_nbclust(Pharmaceuticals_tbl, FUNcluster = hcut, method = "wss")
```



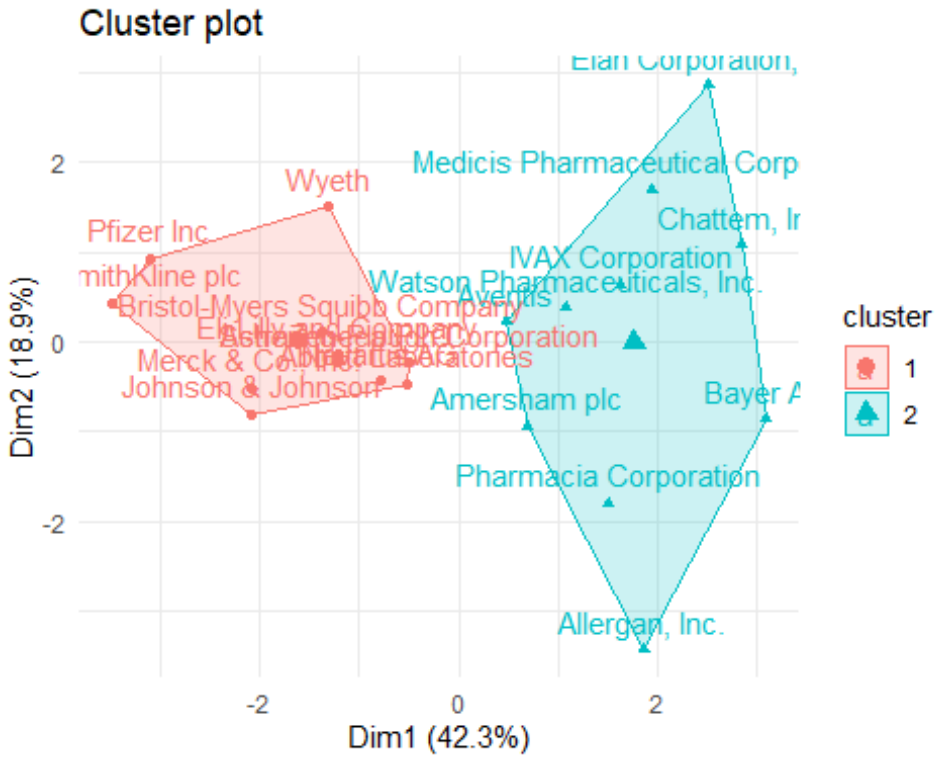
```r
# sillhouette method
fviz_nbclust(Pharmaceuticals_tbl, FUNcluster = hcut, method = "silhouette")
```




```r
# cutree function
cl_1 <- cutree(hclust_1, k = 2)

# table function check the number of pharmaceutical companies in each cluster
table(cl_1)

## cl_1
##  1  2
## 11 10

plot(hclust_1, cex = 0.6)
rect.hclust(hclust_1, k = 2, border= 2:5)
```



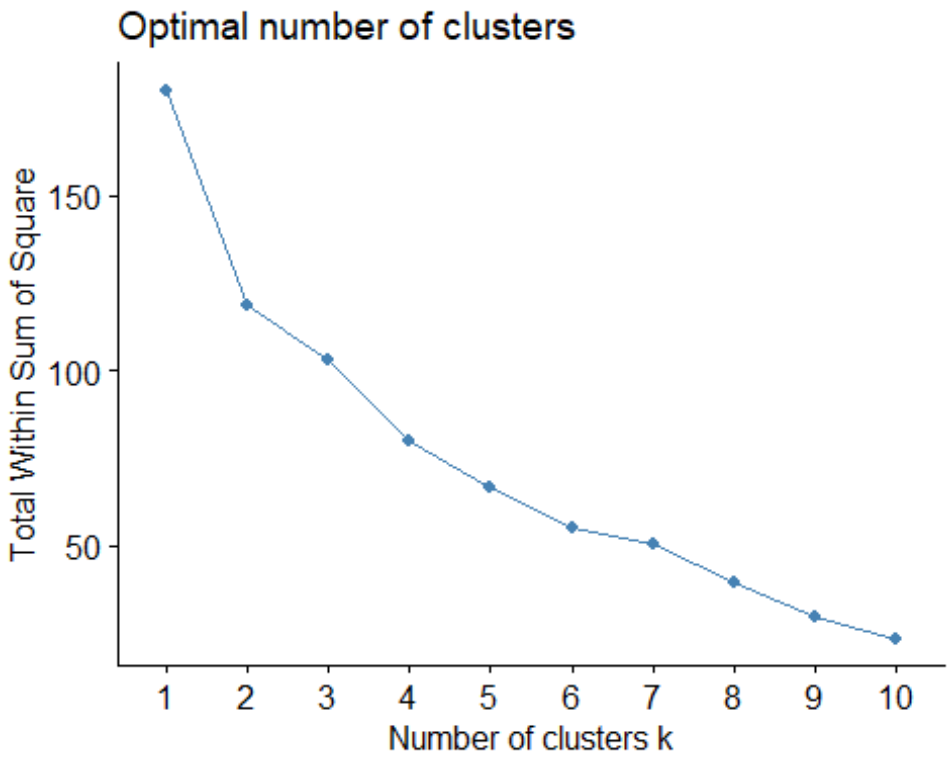

```
# fviz_cluster function to visualize the clusters
fviz_cluster(list(data = Pharmaceuticals_tbl, cluster = cl_1, repel = TRUE))
+
theme_minimal()
```



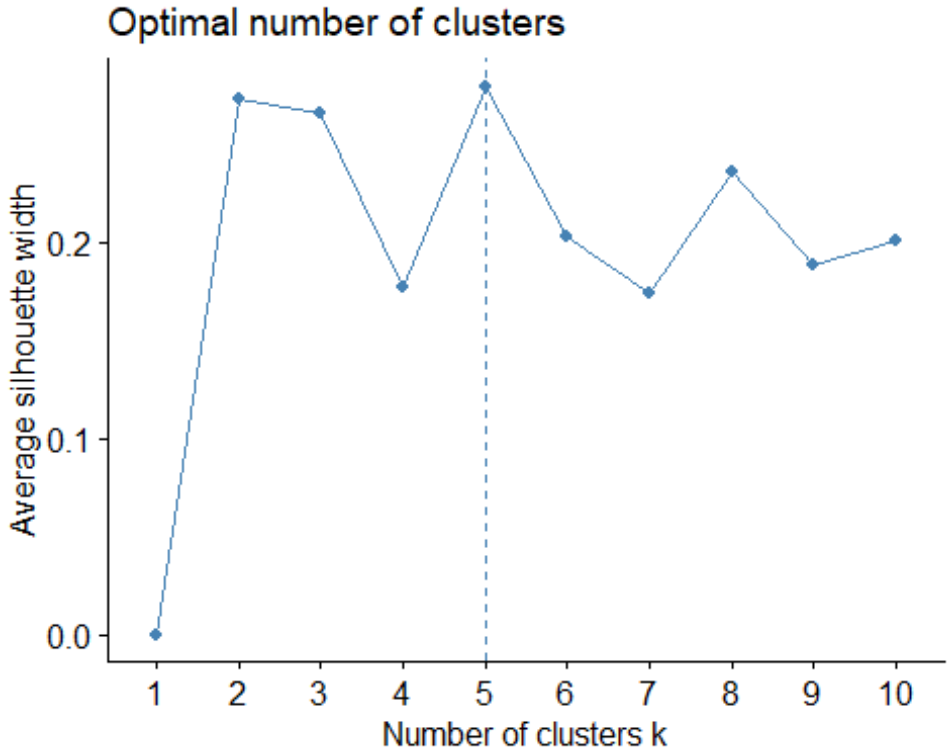

```
# create cluster variable
Pharmaceuticals_tbl$cluster <- cl_1

# aggregate by cluster our variables
Pharmaceuticals_tbl %>%
group_by(cluster) %>%
summarise_all(mean)
```

```
## # A tibble: 2 x 10
##   cluster Market_Cap   Beta PE_Ratio    ROE    ROA Asset_Turnover Leverage
##     <int>      <dbl>  <dbl>    <dbl>  <dbl>  <dbl>          <dbl>    <dbl>
## 1       1      0.673 -0.359   -0.276  0.657  0.834          0.461   -0.333
## 2       2     -0.741  0.395    0.304 -0.722 -0.918         -0.507    0.366
## # ... with 2 more variables: Rev_Growth <dbl>, Net_Profit_Margin <dbl>
```

```
## K-Means Cluster Analysis
# use a data frame only with numeric values and scale the variables because
they were measured in different scales
Pharmaceuticals_tbl <- na.omit(Pharmaceuticals) %>%
dplyr::select(-c(1, 12, 13, 14)) %>%
column_to_rownames(var = "Name") %>%
scale(.) %>% # standardize the values
as.data.frame() # convert to data frame

# elbow method
fviz_nbclust(Pharmaceuticals_tbl, FUNcluster = kmeans, method = "wss")
```




```
# sillhouette method
fviz_nbclust(Pharmaceuticals_tbl, FUNcluster = kmeans, method = "silhouette")
```



```r
# build algorithm
set.seed(88)
k_cluster2 <- kmeans(Pharmaceuticals_tbl, centers = 2, nstart = 50,
iter.max = 10) # k equals 2 clusters

table(k_cluster2$cluster)

##
##  1  2
## 11 10

# check total within and between sum of squares
glance(k_cluster2)

## # A tibble: 1 x 4
##   totss tot.withinss betweenss  iter
##   <dbl>        <dbl>     <dbl> <int>
## 1   180         119.      61.4     1

# dunn index
dunn_k2 <- dunn(clusters = k_cluster2$cluster, Data = Pharmaceuticals_tbl)
dunn_k2

## [1] 0.2546142

set.seed(88)
k_cluster3 <- kmeans(Pharmaceuticals_tbl, centers = 3, nstart = 50,
iter.max = 10) # centers equals 3 clusters



table(k_cluster3$cluster)

##
##  1  2  3
##  4 11  6

# check wSS and BSS
glance(k_cluster3)

## # A tibble: 1 x 4
##   totss tot.withinss betweenss  iter
##   <dbl>        <dbl>     <dbl> <int>
## 1   180         96.0      84.0     2

tidy(k_cluster3)

## # A tibble: 3 x 12
##   Market_Cap   Beta PE_Ratio    ROE    ROA Asset_Turnover Leverage
Rev_Growth
##        <dbl>  <dbl>    <dbl>  <dbl>  <dbl>          <dbl>    <dbl>
<dbl>
```

```
## 1     -0.613  0.270    1.31  -0.961 -1.02           0.231   -0.359     -
0.576
## 2      0.673 -0.359   -0.276  0.657  0.834          0.461   -0.333     -
0.290
## 3     -0.826  0.478   -0.370 -0.563 -0.851         -0.999    0.850
0.916
## # ... with 4 more variables: Net_Profit_Margin <dbl>, size <int>,
## #   withinss <dbl>, cluster <fct>
```

```
# check dunn index
dunn_k3 <- dunn(clusters = k_cluster3$cluster, Data = Pharmaceuticals_tbl)
dunn_k3
```

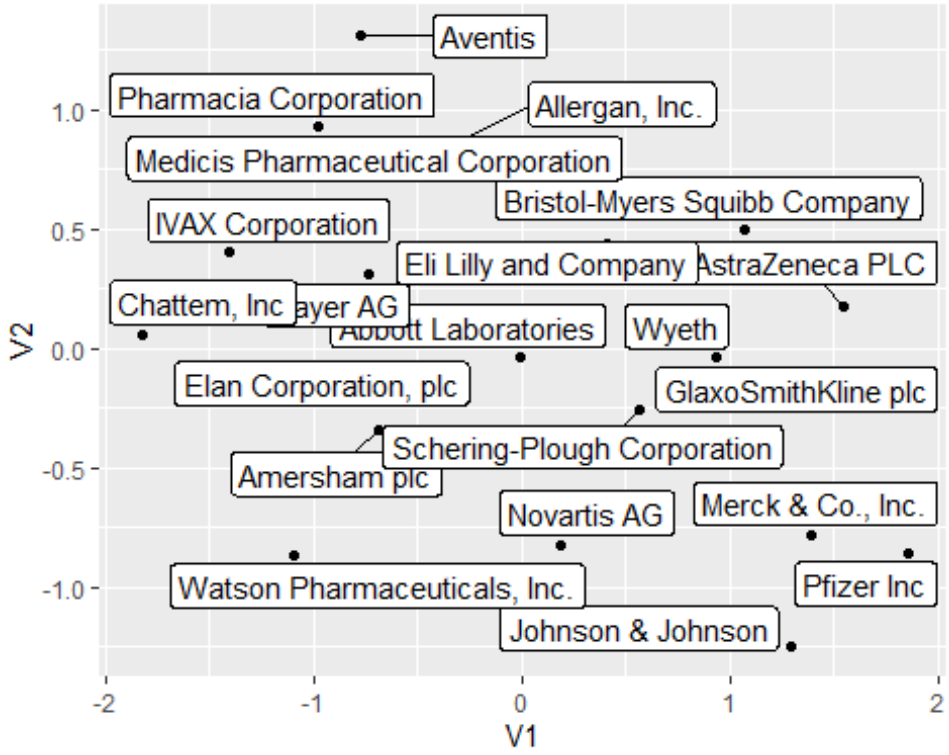
```
## [1] 0.3076927
```

```
# umap our data frame
umap_pharma <- Pharmaceuticals_tbl %>%
umap()

# create umap dataframe
umap_obj <- umap_pharma$layout %>%
as.data.frame() %>%
rownames_to_column(var = "Pharma")

umap_obj
```

```
##                                Pharma           V1          V2
## 1                 Abbott Laboratories -0.006386719 -0.03820645
## 2                     Allergan, Inc. -0.349650540  0.84751259
## 3                       Amersham plc -0.686937893 -0.33873418
## 4                    AstraZeneca PLC  1.544959585  0.17751422
## 5                            Aventis -0.776284656  1.31215800
## 6                          Bayer AG -0.739475679  0.31308901
## 7        Bristol-Myers Squibb Company  1.066758056  0.49523568
## 8                        Chattem, Inc -1.820388156  0.05457267
## 9               Elan Corporation, plc -1.379141023 -0.21700206
## 10              Eli Lilly and Company  0.413959224  0.43995254
## 11                GlaxoSmithKline plc  1.820819183 -0.23503525
## 12                   IVAX Corporation -1.407671528  0.40263602
## 13                  Johnson & Johnson  1.296207766 -1.24741949
## 14 Medicis Pharmaceutical Corporation -1.839490376  0.72966755
## 15                  Merck & Co., Inc.  1.388129673 -0.78318952
## 16                        Novartis AG  0.182985861 -0.82208708
## 17                         Pfizer Inc  1.855880824 -0.85516483
## 18              Pharmacia Corporation -0.977455532  0.92735588
## 19        Schering-Plough Corporation  0.568516501 -0.25933814
## 20       Watson Pharmaceuticals, Inc. -1.089027896 -0.86386390
## 21                             Wyeth  0.933693325 -0.03965325
```

```
# visualize umap dataframe
umap_obj %>%
```

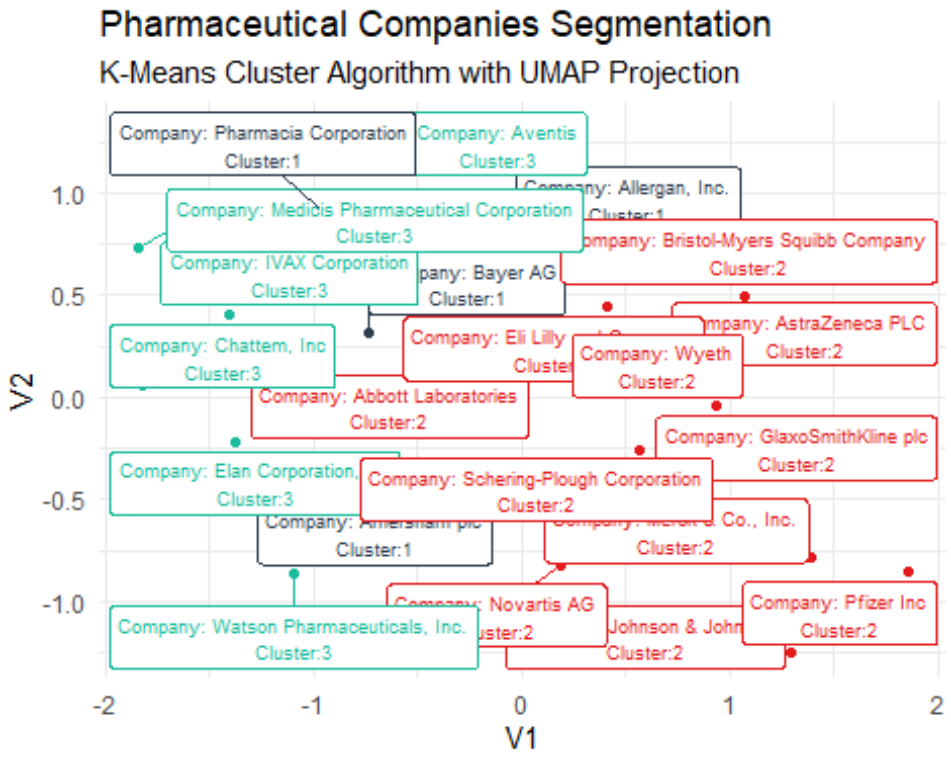
```
  ggplot(aes(V1, V2)) +
  geom_point() +
  geom_label_repel(aes(label = Pharma))
```




```
# use augment to assign the clusters to our pharmaceutical companies
kmeans_tbl <- augment(k_cluster3, Pharmaceuticals_tbl) %>%
  dplyr::select(pharma = .rownames, .cluster)

# join the kmeans data frame with the umap object
kmeans_umap <- kmeans_tbl %>%
  left_join(umap_obj, by = c("pharma" = "Pharma"))

kmeans_umap
```

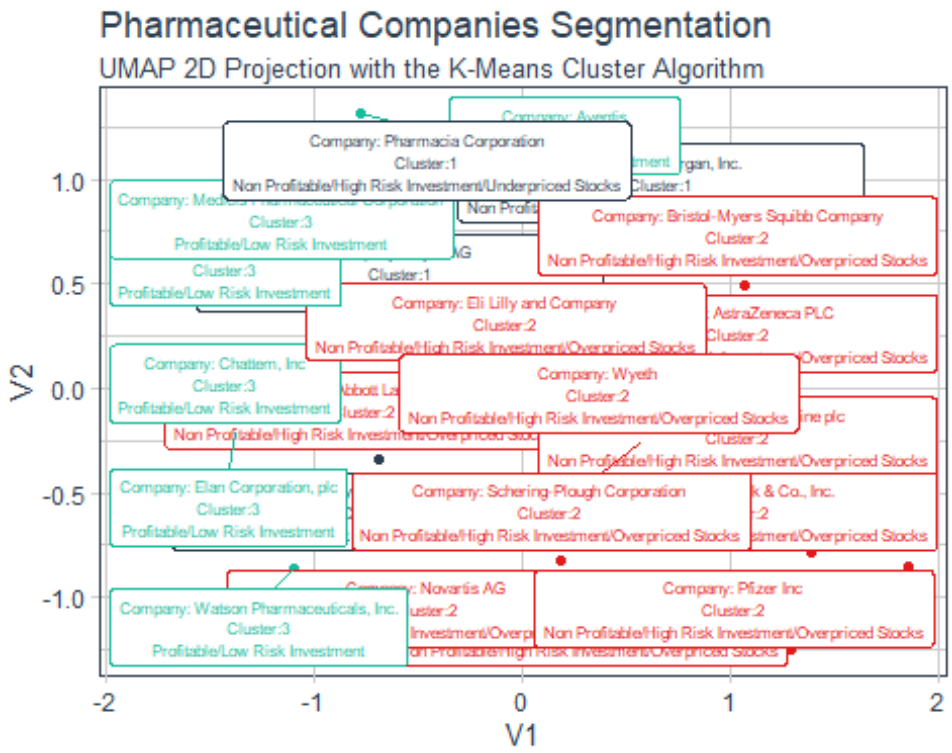
```
## # A tibble: 21 x 4
##    pharma                      .cluster       V1      V2
##    <chr>                       <fct>       <dbl>   <dbl>
##  1 Abbott Laboratories         2         -0.00639 -0.0382
##  2 Allergan, Inc.              1         -0.350    0.848
##  3 Amersham plc                1         -0.687   -0.339
##  4 AstraZeneca PLC             2          1.54     0.178
##  5 Aventis                     3         -0.776    1.31
##  6 Bayer AG                    1         -0.739    0.313
##  7 Bristol-Myers Squibb Company 2         1.07     0.495
##  8 Chattem, Inc                3         -1.82     0.0546
##  9 Elan Corporation, plc       3         -1.38    -0.217
```

```
## 10 Eli Lilly and Company        2         0.414    0.440
## # ... with 11 more rows

kmeans_umap %>%
mutate(label_pharma = str_glue("Company: {pharma}
Cluster:{.cluster}")) %>%
ggplot(aes(V1, V2, color = .cluster)) +
geom_point() +
geom_label_repel(aes(label = label_pharma), size = 2.5) +
guides(color = "none") +
theme_minimal() +
scale_color_tq() +
labs(title = "Pharmaceutical Companies Segmentation",
subtitle = "K-Means Cluster Algorithm with UMAP Projection")
```




```
k_cluster3 %>%
  augment(Pharmaceuticals_tbl) %>%
  dplyr::select(-.rownames) %>%
  group_by(.cluster) %>%
  summarise_all(mean)

## # A tibble: 3 x 10
##   .cluster Market_Cap   Beta PE_Ratio    ROE    ROA Asset_Turnover
Leverage
##   <fct>         <dbl>  <dbl>    <dbl>  <dbl>  <dbl>          <dbl>
<dbl>
## 1 1           -0.613  0.270    1.31  -0.961 -1.02           0.231   -
```

```
0.359
## 2 2             0.673 -0.359   -0.276  0.657  0.834          0.461   -
0.333
## 3 3            -0.826  0.478   -0.370 -0.563 -0.851         -0.999
0.850
## # ... with 2 more variables: Rev_Growth <dbl>, Net_Profit_Margin <dbl>

# create tibble withthe characteristics of the 3 cluster
cluster_tibble <- tibble::tribble(~.cluster, ~cluster.label,
                                  1, "Non Profitable/High Risk
Investment/Underpriced Stocks",
                                  2, "Non Profitable/High Risk
Investment/Overpriced Stocks",
                                  3, "Profitable/Low Risk Investment")


# make .cluster variable a factor
cluster_tibble <- cluster_tibble %>%
  mutate(.cluster = as.factor(as.character(.cluster)))

# clusters visualization
kmeans_umap %>%
  left_join(cluster_tibble) %>%
  mutate(label_pharma = str_glue("Company: {pharma}
                                 Cluster:{.cluster}
                                 {cluster.label}")) %>%
  ggplot(aes(V1, V2, color = .cluster)) +
  geom_point() +
  geom_label_repel(aes(label = label_pharma), size = 2) +
  guides(color = "none") +
  theme_tq() +
  scale_color_tq() +
  labs(title = "Pharmaceutical Companies Segmentation",
       subtitle = "UMAP 2D Projection with the K-Means Cluster Algorithm")

## Joining, by = ".cluster"
```


Pharmaceutical Companies Segmentation
UMAP 2D Projection with the K-Means Cluster Algorithm
Company: Pharmacia Corporation
Cluster:1
Non Profitable/High Risk Investment/Underpriced Stocks
Company: Bristol-Myers Squibb Company
Cluster:2
Non Profitable/High Risk Investment/Overpriced Stocks
Company: Eli Lilly and Company
Cluster:2
Company: Wyeth
Cluster:2
Non Profitable/High Risk Investment/Overpriced Stocks
Company: Schering-Plough Corporation
Cluster:2
Non Profitable/High Risk Investment/Overpriced Stocks
Company: Pfizer Inc
Cluster:2
Non Profitable/High Risk Investment/Overpriced Stocks
Company: Chattem, Inc.
Cluster:3
Profitable/Low Risk Investment
Company: Elan Corporation, plc
Cluster:3
Profitable/Low Risk Investment
Company: Watson Pharmaceuticals, Inc.
Cluster:3
Profitable/Low Risk Investment
V1
V2
-2
-1
0
1
2
1.0
0.5
0.0
-0.5
-1.0