K-means Clustering Algorithm

Melissa Paniagua

10/21/2020

Assignment 4 | Module 6

The purpose of this assignment is to use k-Means for clustering.

For this project, we are going to use a dataset containing information on 1302 American colleges and universities offering an undergraduate program. For each university, there are 20 measurements, which are the following:

- College Name: University's name.
- State: Location.
- Private / Public : Priviate is 1 and Public is 2.
- appli. rec'd: Number of applications received.
- appl. accepted: Number of applicants accepted.
- new stud. enrolled: Number of students enrolled.
- new stud. from top 10%: New students from top 10%.
- new stud. from top 25%: New students from top 25%.
- FT undergrad: Number of full time students.
- PT undergrad: Number of part-time students.
- in-state tuition: Amount of tuition in-state.
- out-of-state tuition: Amount of tuition out-of-state.
- room: Number of rooms.
- board: Number of boards
- add. fees: Extra fees.
- estim. book costs: Amount spent on books.
- estim. personal: Amount spent on personal expences.
- % fac. w/PHD: Percentage of faculty with doctorate degree.
- stud./fac. ratio: Ratio of the number of students and the faculty.
- Graduation rate: Rate of people that finish the program.

```r
#Load the libraries needed
library(readr)
library(tidyverse)
library(factoextra)
library(psych)
library(ggplot2)
library(ggpubr)
library(corrplot)
library(RColorBrewer)
library(data.table)
library(caret)

# Load the file
Universities <- read_csv("Universities.csv")

#Show the first 6 rows and 4 first variables
head(Universities)
```

```
# A tibble: 6 x 20
  `College Name` State `Public (1)/ Pr~ `# appli. rec'd` `# appl. accept~
  <chr>          <chr>            <dbl>            <dbl>            <dbl>
1 Alaska Pacifi~ AK                   2              193              146
2 University of~ AK                   1             1852             1427
3 University of~ AK                   1              146              117
4 University of~ AK                   1             2065             1598
5 Alabama Agri.~ AL                   1             2817             1920
6 Faulkner Univ~ AL                   2              345              320
# ... with 15 more variables: `# new stud. enrolled` <dbl>, `% new stud. from
#   top 10%` <dbl>, `% new stud. from top 25%` <dbl>, `# FT undergrad` <dbl>,
#   `# PT undergrad` <dbl>, `in-state tuition` <dbl>, `out-of-state
#   tuition` <dbl>, room <dbl>, board <dbl>, `add. fees` <dbl>, `estim. book
#   costs` <dbl>, `estim. personal $` <dbl>, `% fac. w/PHD` <dbl>, `stud./fac.
#   ratio` <dbl>, `Graduation rate` <dbl>
```

```r
#Show the last 6 rows and 4 first variables
tail(Universities)
```

```
# A tibble: 6 x 20
  `College Name` State `Public (1)/ Pr~ `# appli. rec'd` `# appl. accept~
  <chr>          <chr>            <dbl>            <dbl>            <dbl>
1 West Virginia~ WV                   1             1594             1572
2 West Virginia~ WV                   1             1869               NA
3 West Virginia~ WV                   1             9630             7801
4 West Virginia~ WV                   2             1566             1400
5 Wheeling Jesu~ WV                   2              903              755
6 University of~ WY                   1             2029             1516
# ... with 15 more variables: `# new stud. enrolled` <dbl>, `% new stud. from
#   top 10%` <dbl>, `% new stud. from top 25%` <dbl>, `# FT undergrad` <dbl>,
#   `# PT undergrad` <dbl>, `in-state tuition` <dbl>, `out-of-state
#   tuition` <dbl>, room <dbl>, board <dbl>, `add. fees` <dbl>, `estim. book
#   costs` <dbl>, `estim. personal $` <dbl>, `% fac. w/PHD` <dbl>, `stud./fac.
#   ratio` <dbl>, `Graduation rate` <dbl>
```

It is important to run the head and tail of the dataframe to confirm that the dataframe is similar among its data points.

Data Exploration

```
# To get the total number of rows and columns
dim(Universities)
```

```
[1] 1302   20
```

This output shows that the University data frame has 1302 data points and 20 variables.

```
# See the data frame structure
str(Universities)
```

```
tibble [1,302 x 20] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ College Name            : chr [1:1302] "Alaska Pacific University" "University of Alaska at Fairbanks
 $ State                   : chr [1:1302] "AK" "AK" "AK" "AK" ...
 $ Public (1)/ Private (2) : num [1:1302] 2 1 1 1 1 2 1 1 1 2 ...
 $ # appli. rec'd          : num [1:1302] 193 1852 146 2065 2817 ...
 $ # appl. accepted        : num [1:1302] 146 1427 117 1598 1920 ...
 $ # new stud. enrolled    : num [1:1302] 55 928 89 1162 984 ...
 $ % new stud. from top 10%: num [1:1302] 16 NA 4 NA NA NA 18 NA 25 67 ...
 $ % new stud. from top 25%: num [1:1302] 44 NA 24 NA NA 27 78 NA 57 88 ...
 $ # FT undergrad          : num [1:1302] 249 3885 492 6209 3958 ...
 $ # PT undergrad          : num [1:1302] 869 4519 1849 10537 305 ...
 $ in-state tuition        : num [1:1302] 7560 1742 1742 1742 1700 ...
 $ out-of-state tuition    : num [1:1302] 7560 5226 5226 5226 3400 ...
 $ room                    : num [1:1302] 1620 1800 2514 2600 1108 ...
 $ board                   : num [1:1302] 2500 1790 2250 2520 1442 ...
 $ add. fees               : num [1:1302] 130 155 34 114 155 300 124 84 NA 120 ...
 $ estim. book costs       : num [1:1302] 800 650 500 580 500 350 300 500 600 400 ...
 $ estim. personal $       : num [1:1302] 1500 2304 1162 1260 850 ...
 $ % fac. w/PHD            : num [1:1302] 76 67 39 48 53 52 72 48 85 74 ...
 $ stud./fac. ratio        : num [1:1302] 11.9 10 9.5 13.7 14.3 32.8 18.9 18.7 16.7 14 ...
 $ Graduation rate         : num [1:1302] 15 NA 39 NA 40 55 51 15 69 72 ...
 - attr(*, "spec")=
  .. cols(
  ..   `College Name` = col_character(),
  ..   State = col_character(),
  ..   `Public (1)/ Private (2)` = col_double(),
  ..   `# appli. rec'd` = col_double(),
  ..   `# appl. accepted` = col_double(),
  ..   `# new stud. enrolled` = col_double(),
  ..   `% new stud. from top 10%` = col_double(),
  ..   `% new stud. from top 25%` = col_double(),
  ..   `# FT undergrad` = col_double(),
  ..   `# PT undergrad` = col_double(),
  ..   `in-state tuition` = col_double(),
  ..   `out-of-state tuition` = col_double(),
```

```
..   room = col_double(),
..   board = col_double(),
..   `add. fees` = col_double(),
..   `estim. book costs` = col_double(),
..   `estim. personal $` = col_double(),
..   `% fac. w/PHD` = col_double(),
..   `stud./fac. ratio` = col_double(),
..   `Graduation rate` = col_double()
.. )
```

This data frame has 17 numerical variables such as tuition, graduation rate, etc. and three categorical variables, which are the following:

- University's name
- State
- Private or Public. It is important to identify that even though this variable is recognized by R Studio as numeric (0's and 1's), the goal is to determine whether it is a private or public school. So, this variable will be treated as categorical.

```
# To get descriptive statistics
summary(Universities)
```

```
 College Name          State           Public (1)/ Private (2)
 Length:1302        Length:1302        Min.   :1.000
 Class :character   Class :character   1st Qu.:1.000
 Mode  :character   Mode  :character   Median :2.000
                                       Mean   :1.639
                                       3rd Qu.:2.000
                                       Max.   :2.000

 # appli. rec'd    # appl. accepted  # new stud. enrolled
 Min.   :   35.0   Min.   :   35.0   Min.   :  18.0
 1st Qu.:  695.8   1st Qu.:  554.5   1st Qu.: 236.0
 Median : 1470.0   Median : 1095.0   Median : 447.0
 Mean   : 2752.1   Mean   : 1870.7   Mean   : 778.9
 3rd Qu.: 3314.2   3rd Qu.: 2303.0   3rd Qu.: 984.0
 Max.   :48094.0   Max.   :26330.0   Max.   :7425.0
 NA's   :10        NA's   :11        NA's   :5
 % new stud. from top 10% % new stud. from top 25% # FT undergrad
 Min.   : 1.00            Min.   :  6.00           Min.   :   59
 1st Qu.:13.00            1st Qu.: 36.75           1st Qu.:  966
 Median :21.00            Median : 50.00           Median : 1812
 Mean   :25.67            Mean   : 52.35           Mean   : 3693
 3rd Qu.:32.00            3rd Qu.: 66.00           3rd Qu.: 4540
 Max.   :98.00            Max.   :100.00           Max.   :31643
 NA's   :235              NA's   :202              NA's   :3
 # PT undergrad    in-state tuition out-of-state tuition      room
 Min.   :    1.0   Min.   :  480    Min.   : 1044        Min.   : 500
 1st Qu.:  131.2   1st Qu.: 2580    1st Qu.: 6111        1st Qu.:1710
 Median :  472.0   Median : 8050    Median : 8670        Median :2200
 Mean   : 1081.5   Mean   : 7897    Mean   : 9277        Mean   :2515
 3rd Qu.: 1313.0   3rd Qu.:11600    3rd Qu.:11659        3rd Qu.:3040
```

```
 Max.   :21836.0   Max.   :25750    Max.   :25750        Max.   :7400
 NA's   :32        NA's   :30       NA's   :20           NA's   :321
     board        add. fees      estim. book costs estim. personal $
 Min.   : 531   Min.   :   9.0   Min.   :  90      Min.   :  75
 1st Qu.:1619   1st Qu.: 130.0   1st Qu.: 480      1st Qu.: 900
 Median :1980   Median : 264.5   Median : 502      Median :1250
 Mean   :2061   Mean   : 392.0   Mean   : 550      Mean   :1389
 3rd Qu.:2402   3rd Qu.: 480.0   3rd Qu.: 600      3rd Qu.:1794
 Max.   :6250   Max.   :4374.0   Max.   :2340      Max.   :6900
 NA's   :498    NA's   :274      NA's   :48        NA's   :181
  % fac. w/PHD    stud./fac. ratio Graduation rate
 Min.   :  8.00   Min.   : 2.30    Min.   :  8.00
 1st Qu.: 57.00   1st Qu.:11.80    1st Qu.: 47.00
 Median : 71.00   Median :14.30    Median : 60.00
 Mean   : 68.65   Mean   :14.86    Mean   : 60.41
 3rd Qu.: 82.00   3rd Qu.:17.60    3rd Qu.: 74.00
 Max.   :105.00   Max.   :91.80    Max.   :118.00
 NA's   :32       NA's   :2        NA's   :98
```

This help us to see some descriptive statistics and also to determine that all variables have missing values, but the "Private and Public" variable.

Data Prepararation

```
# Rename column names
Universities <- rename_with(Universities, ~ tolower(gsub(" ", ".", .x, fixed = TRUE)))
Universities <- rename_with(Universities, ~ tolower(gsub("#", "num", .x, fixed = TRUE)))
Universities <- rename_with(Universities, ~ tolower(gsub("$", "", .x, fixed = TRUE)))
Universities <- rename_with(Universities, ~ tolower(gsub("%", "pc", .x, fixed = TRUE)))
Universities <- rename_with(Universities, ~ tolower(gsub("..", ".", .x, fixed = TRUE)))
```

Remove the obervation to predict (Tufts University)

It is important to remove the Tufts University observation before normalize the data frame because this data point will be used to predict which cluster it will belong to. If we add this data point on the normalization step, it will add some bias to the prediction and the distance.

```
# Remove the Tufts University
mydf_notufts <- subset(Universities, college.name!= 'Tufts University' )

# Do prove we remove it
dim(mydf_notufts)
```

```
[1] 1301   20
```

Our data frame changed from 1032 observations to 1301.

Normalization or Scaling

As we have seen in previous models, it is essential to normalize the data first in order to avoid assumptions based on the distribution of the data and different units of the variables.

```
# Select only numerical variables
mydf_numerical <- mydf_notufts[, 4:20]

# Normalize the data using the scale function (z-score)
mydf_numerical <- scale(mydf_numerical)

#To see the first 6 rows and the first three variables
head(mydf_numerical)[1:6, 1:3]
```

```
     num.appli.rec'd num.appl.accepted num.new.stud.enrolled
[1,]     -0.72169040        -0.7655130            -0.8177200
[2,]     -0.25314674        -0.1964884             0.1688986
[3,]     -0.73496439        -0.7783949            -0.7792949
[4,]     -0.19299015        -0.1205296             0.4333530
[5,]      0.01939372         0.0225039             0.2321868
[6,]     -0.67876175        -0.6882216            -0.6775817
```

Here we can see our numerical data variables have been normalized.

Remove missing values

Remember that normalizing the data first and then removing the missing values will help us to find the true mean and true standard deviation to accurately run the model.

Now, it is essential to remove the missing values to be able to run the K-means model. If we do not do this step, the K-means function will return an error.

```
# Remove missing values
mydf_numerical <- na.omit(mydf_numerical)

# To get the total number of rows and columns
dim(mydf_numerical)
```

```
[1] 471  17
```

Here we can see that 831 rows were removed.

Computing the Distance

For computing the distance, we are going to use the get_dist function from the factoextra package in R.

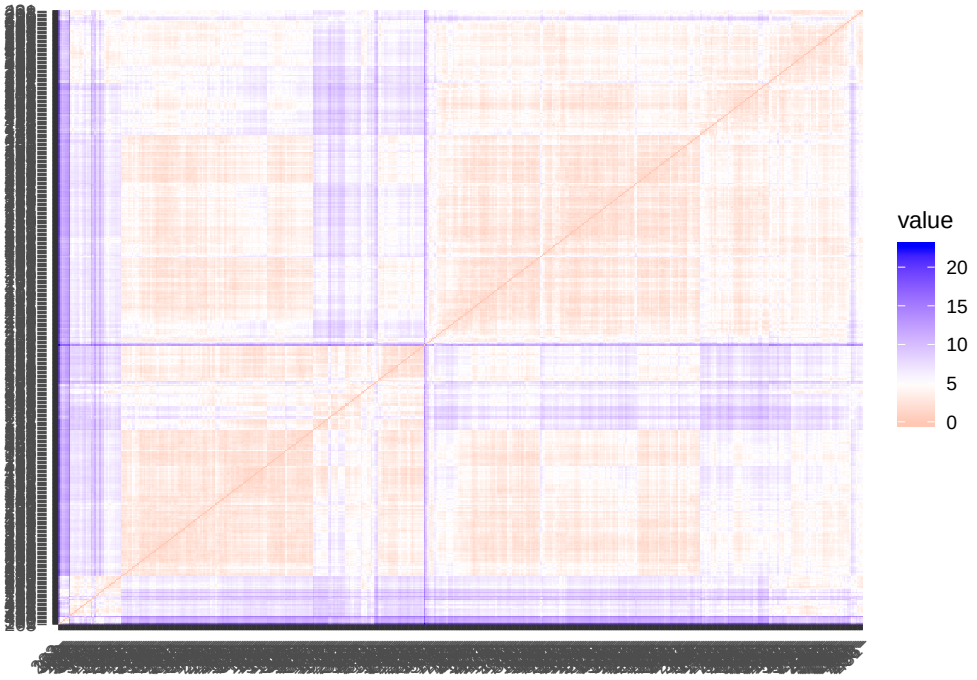
The get_dist() function computes a distance matrix between the rows of a data matrix, and it uses the Euclidean distance as default.

```
# Computing the distance. Euclidean distance calculated by default
distance <- get_dist(mydf_numerical)

#To see the first 6 rows
head(distance)
```

```
[1] 3.683413 5.477225 2.659396 3.465611 4.059029 4.014461
```

```r
# Let's visualize our distances. The fviz_dist() function visualizes a distance matrix
fviz_dist(distance)
```



This graph is a distance matrix. As we can see, the diagonal values are zeros because it is showing the distance between any point againts itself. The purple represents the furthest distance between any pair of observations.

Running the K-means Model

In this part, we are going to run the K-means model using a random k number. In this case, we randomly choose k=2, and we are going to perform 30 iterations.

```r
# To maintain same values
set.seed(123)

# To run the kmeans model
k2 <- kmeans(mydf_numerical, centers = 2, nstart = 30)

# To see the results
print(k2)
```

```
K-means clustering with 2 clusters of sizes 412, 59

Cluster means:
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1      -0.1728961        -0.2237562            -0.3166533
```

```
2       2.1068713         2.2490850             2.2306319
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                 0.1206778                 0.1279823       -0.3524606
2                 0.1929760                 0.3830006        2.2346933
  num.pt.undergrad in-state.tuition out-of-state.tuition       room       board
1       -0.3667812        0.4309360            0.3843378 -0.2515195  0.11777197
2        1.2050979       -0.7389398           -0.1782070 -0.2755424 -0.07364994
     add.fees estim.book.costs estim.personal. pc.fac.w/phd stud./fac.ratio
1 -0.08331224       -0.0242605      -0.2163737    0.1742609      -0.2326920
2  0.36275858        0.1145987       0.6420277    0.8386122       0.2402606
  graduation.rate
1      0.30512980
2      0.06163317

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1
 [75] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1
[112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
[149] 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1
[186] 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1
[223] 1 1 1 1 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
[260] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1
[297] 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1 1 2 1
[334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2
[371] 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[408] 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 1 1 1 1 1 1 1 1 2
[445] 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1

Within cluster sum of squares by cluster:
[1] 4525.881 1220.660
 (between_SS / total_SS =  21.5 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

This chart shows 2 clusters of sizes 412, 59. We also see the clusters means of each variable based on each cluster, and how each data point is assigned. For example the first row was assigned to cluster 1, etc.

```
# To output the centers or centroids
k2$centers
```

```
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1      -0.1728961        -0.2237562            -0.3166533
2       2.1068713         2.2490850             2.2306319
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                 0.1206778                 0.1279823       -0.3524606
2                 0.1929760                 0.3830006        2.2346933
  num.pt.undergrad in-state.tuition out-of-state.tuition       room       board
1       -0.3667812        0.4309360            0.3843378 -0.2515195  0.11777197
2        1.2050979       -0.7389398           -0.1782070 -0.2755424 -0.07364994
     add.fees estim.book.costs estim.personal. pc.fac.w/phd stud./fac.ratio
```

```
1 -0.08331224       -0.0242605       -0.2163737     0.1742609       -0.2326920
2  0.36275858        0.1145987        0.6420277     0.8386122        0.2402606
  graduation.rate
1      0.30512980
2      0.06163317
```

```r
# To output the cluster's size.
k2$size
```

```
[1] 412  59
```

```r
# Let's run table function to identify to which cluster those sizes belong
table(k2$cluster)
```

```
  1   2
412  59
```

Here we can see 412 universities belong to cluster 1, and 59 universities to cluster 2.

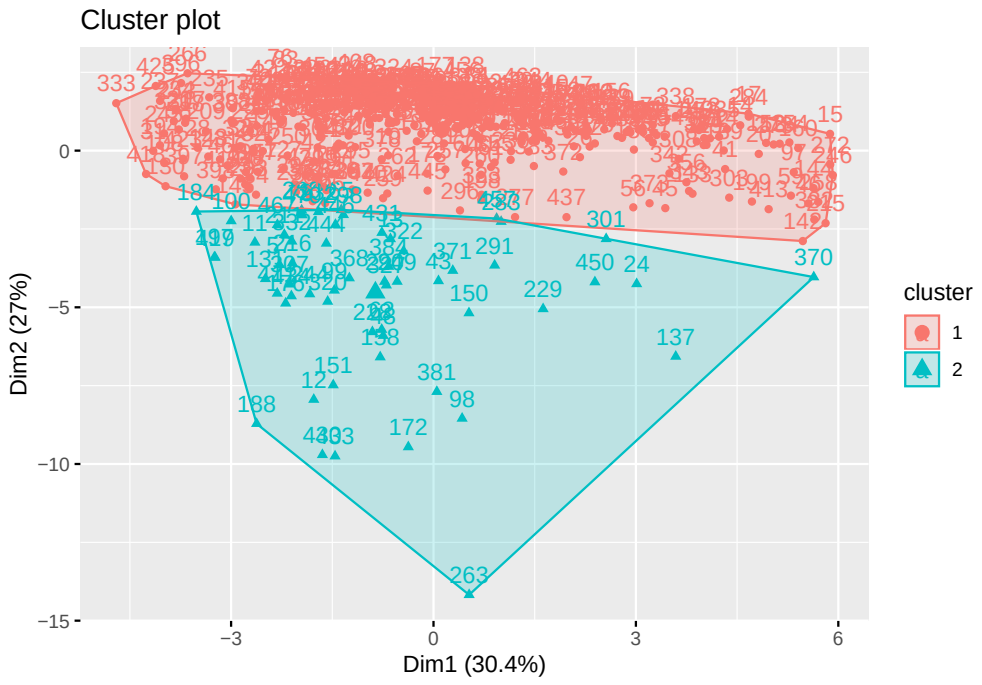
```r
# Identify the cluster of the 100th observation as an example
k2$cluster[100]
```

```
[1] 2
```

This output shows that the observation 100th has been classified to cluster 2.

Visualize

```r
# To visualize the output
fviz_cluster(k2, data = mydf_numerical)
```



This visual helps us to see how the Kmeans model has grouped our universities based on their numerical variables in two groups. But, remember that we chose the number of groups by selecting k=2 without strong grounds. For this reason, now we are going to run two methods to choose our optimal k.

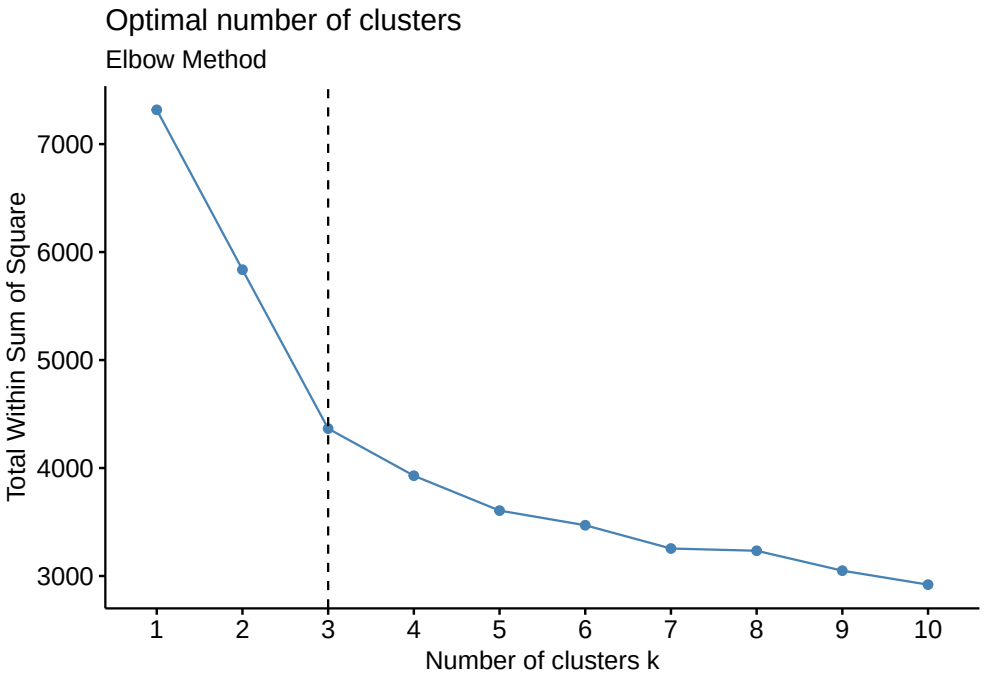
Choosing the best K

There are different ways of choosing the best K. Some are domain knowledge, gap statistic method, and also a NbClust() function. For our purposes, we are going to use two graphical methods:

- The Elbow Method
- The Silhouette Method

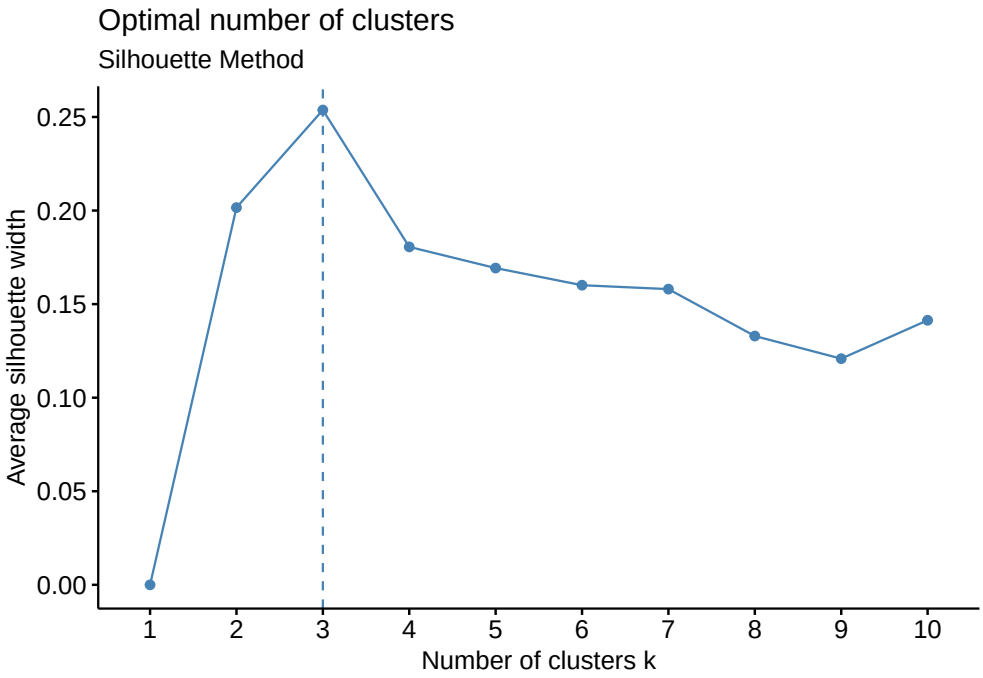
Elbow Method

```
# Visualizing the Elbow Method
fviz_nbclust(mydf_numerical, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2) + # add line. However, we have input the value manually.
  labs(subtitle = "Elbow Method") # add the subtitle
```



Silhouette Method

```
# Visualizing the Silhouette Method
fviz_nbclust(mydf_numerical, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette Method") # add the subtitle
```



These two last graphs are helping us to choose the best k to use in the K-means model. As we can see, both methods determined that the best k is 3.

So, let's now run the kmeans model again using the optimal k = 3

K-means Model using the best k

We are going to run the k-means model using the optimal k. As mentioned before, the Elbow method and Silhouette method determined k=3, and we are going to perform 30 iterations.

```
# To maintain same values
set.seed(123)

# To run the kmeans model
k3 <- kmeans(mydf_numerical, centers = 3, nstart = 30)

# To see the results
print(k3)
```

```
K-means clustering with 3 clusters of sizes 46, 276, 149

Cluster means:
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1       2.3924543        2.56607076             2.5322330
2      -0.2964704       -0.29561980            -0.3211226
3       0.1667458        0.02725002            -0.1792399
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                 0.2645288                 0.4079769        2.5625490
2                -0.3897501                -0.3560933       -0.3275416
```

```
3                1.0503862                 1.0391984       -0.2741115
  num.pt.undergrad in-state.tuition out-of-state.tuition       room      board
1        1.4466178       -0.8004011           -0.1956077 -0.2785843 -0.0558225
2       -0.2732909       -0.1278216           -0.2296380 -0.4527726 -0.2079440
3       -0.4773773        1.3828550            1.4779261  0.1201146  0.6989065
     add.fees estim.book.costs estim.personal. pc.fac.w/phd stud./fac.ratio
1  0.34791254       0.15258556     0.786930181    0.8975252      0.28808286
2 -0.06657495      -0.06011004    -0.006974134   -0.2409599      0.01670269
3 -0.07081328       0.04253309    -0.574095557    0.9831703     -0.66815761
  graduation.rate
1      0.03059455
2     -0.13026619
3      1.09997284

Clustering vector:
  [1] 2 2 3 2 2 2 2 2 2 2 1 1 1 3 3 3 3 3 2 3 2 3 3 1 3 3 2 2 3 2 2 2 2 2 3 2 3
 [38] 3 2 2 3 2 1 3 3 3 3 1 2 2 3 2 2 3 3 3 1 2 3 2 2 1 2 2 2 2 2 2 3 2 2 2 3 2
 [75] 2 2 2 2 2 2 2 2 2 3 2 3 2 3 3 3 2 3 2 2 2 2 3 1 1 2 2 3 3 3 2 2 2 2 2 2 2
[112] 3 3 3 3 2 2 2 2 2 2 2 2 3 2 2 2 3 2 2 1 3 3 3 2 3 1 2 3 3 2 3 2 3 2 2 2 2
[149] 3 1 1 3 3 3 3 3 2 1 2 3 3 2 2 2 2 3 3 2 3 3 2 1 2 2 2 1 2 3 3 2 3 3 3 2 2
[186] 2 2 1 2 2 2 2 2 2 2 2 1 2 3 2 2 2 2 2 2 2 2 2 2 2 1 2 2 3 3 1 2 2 2 2 2 2
[223] 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 2 3 2 2 1 2 2 3 2 2 2 2 2 3 2
[260] 2 2 2 1 2 2 2 2 3 2 2 2 3 3 3 3 3 3 3 3 2 3 3 2 2 2 3 3 3 3 2 2 3 1 1 2 2 2 2 3
[297] 2 2 2 2 2 3 3 3 2 2 2 1 3 2 2 2 2 3 2 3 2 3 3 2 3 1 2 1 3 2 3 2 1 2 2 2 2 2 1 2
[334] 2 2 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 3 3 2 3 3 3 2 2 3 3 2 2 2 2 2 2 3 1 2 3
[371] 1 3 3 3 2 2 2 2 2 2 2 1 3 3 1 2 2 2 2 2 2 3 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2
[408] 3 2 2 1 3 3 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 3 3 2 1 1 2 1 1 2 2 3 2 3 2 3 2 2 1
[445] 2 1 2 2 3 1 2 2 2 2 3 2 3 3 3 3 2 3 3 3 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1]  982.7664 2248.4119 1133.9450
 (between_SS / total_SS =  40.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```
# To output the centers or centroids
k3$centers
```

```
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1       2.3924543        2.56607076             2.5322330
2      -0.2964704       -0.29561980            -0.3211226
3       0.1667458        0.02725002            -0.1792399
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                 0.2645288                 0.4079769        2.5625490
2                -0.3897501                -0.3560933       -0.3275416
3                 1.0503862                 1.0391984       -0.2741115
  num.pt.undergrad in-state.tuition out-of-state.tuition       room      board
1        1.4466178       -0.8004011           -0.1956077 -0.2785843 -0.0558225
2       -0.2732909       -0.1278216           -0.2296380 -0.4527726 -0.2079440
3       -0.4773773        1.3828550            1.4779261  0.1201146  0.6989065
     add.fees estim.book.costs estim.personal. pc.fac.w/phd stud./fac.ratio
```

```
1  0.34791254       0.15258556     0.786930181    0.8975252      0.28808286
2 -0.06657495      -0.06011004    -0.006974134   -0.2409599      0.01670269
3 -0.07081328       0.04253309    -0.574095557    0.9831703     -0.66815761
  graduation.rate
1      0.03059455
2     -0.13026619
3      1.09997284
```

```
# To output the cluster's size.
k3$size
```

```
[1]  46 276 149
```

```
# Let's run table function to identify to which cluster those sizes belong
table(k3$cluster)
```

```
  1   2   3
 46 276 149
```

The number of universities on our new clusters are 46 to cluster 1, 276 to cluster 2, and 149 to cluster 3.

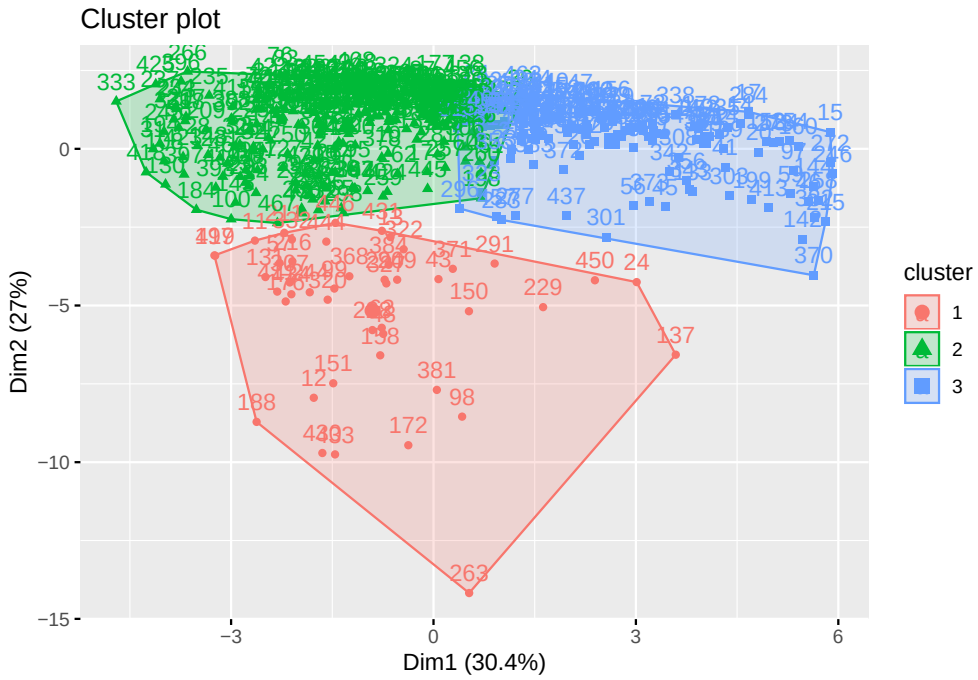
```
# Identify the cluster of the 100th observation as an example
k3$cluster[100]
```

```
[1] 2
```

Our 100th observation will remain in cluster 2.

Visualize using best k

```
# To visualize the output
fviz_cluster(k3, data = mydf_numerical)
```



This visual shows the three clusters developed by the model, which are classified better than the previous one.

Questions

2. How many clusters seem reasonable for describing these data? What was your optimal K?

As we can see in our examples, at the beginning we ran the Kmeans model using a random k. In the beginning, there wasn't a way to describe the optimal number of clusters that best described the data. After performing the Elbow Method and the Silhouette Method, we were able to get the optimal k that best described our University's data set, which the answer is k=3.

3. Compare the summary statistics for each cluster and describe each cluster in this context (e.g., "Universities with high tuition, low acceptance rate. . . ").

First, we will create a new data frame and add a column to identify the clusters by row utilizing unnormalized data. Here, it is essential to go back to our original data to have a better understanding of the clusters.

```
# Create a new df using not normalized data, and add a cluster column.
df_cluster_unnorm <- data.frame(na.omit(Universities)[, 4:20], cluster = as.factor(k3$cluster))

# To show the first 6 rows and last 5 columns
df_cluster_unnorm[1:6, 14:18]
```

```
  estim.personal. pc.fac.w.phd stud..fac.ratio graduation.rate cluster
1            1500           76            11.9              15       2
2            1162           39             9.5              39       2
3             900           74            14.0              72       3
```

```
4            1100           63            11.4              44       2
5            1400           56            15.5              46       2
6            2200           96             6.7              33       2
```

Here we prove that our new data frame has the cluster column.

Now, we are going to the summary statistics, and for that we are going to use the describeBy function.

```
# Summary statistics unnormalized data
describeBy(x = df_cluster_unnorm, group = "cluster", skew = FALSE)
```

```
 Descriptive statistics by group
group: 1
                          vars  n     mean      sd    min     max   range      se
num.appli.rec.d              1 46 11219.43 6890.38 4418.0 48094.0 43676.0 1015.93
num.appl.accepted            2 46  7646.13 3992.30 2737.0 26330.0 23593.0  588.63
num.new.stud.enrolled        3 46  3019.17 1155.14 1567.0  6392.0  4825.0  170.32
pc.new.stud.from.top.10pc    4 46    30.48   15.26   12.0    75.0    63.0    2.25
pc.new.stud.from.top.25pc    5 46    60.83   17.17   29.0    95.0    66.0    2.53
num.ft.undergrad             6 46 15342.67 5582.39 8544.0 31643.0 23099.0  823.08
num.pt.undergrad             7 46  3500.57 3464.31  114.0 21836.0 21722.0  510.79
in.state.tuition             8 46  3613.85 3675.91  672.0 18420.0 17748.0  541.98
out.of.state.tuition         9 46  8454.61 2958.73 4104.0 18420.0 14316.0  436.24
room                        10 46  2193.41  719.78  880.0  3990.0  3110.0  106.13
board                       11 46  2022.98  455.90 1120.0  3435.0  2315.0   67.22
add.fees                    12 46   555.28  519.26   20.0  3247.0  3227.0   76.56
estim.book.costs            13 46   575.48  119.15   96.0   858.0   762.0   17.57
estim.personal.             14 46  1951.91  744.96  600.0  3630.0  3030.0  109.84
pc.fac.w.phd                15 46    84.61    6.23   70.0    93.0    23.0    0.92
stud..fac.ratio             16 46    16.36    4.18    7.8    24.7    16.9    0.62
graduation.rate             17 46    60.96   14.56   34.0    95.0    61.0    2.15
cluster*                    18 46     1.00    0.00    1.0     1.0     0.0    0.00
------------------------------------------------------------------------------------------------------------------------
group: 2
                          vars   n    mean      sd    min     max   range     se
num.appli.rec.d              1 276 1698.60 1665.38   77.0  8399.0  8322.0 100.24
num.appl.accepted            2 276 1203.83 1040.33   61.0  5027.0  4966.0  62.62
num.new.stud.enrolled        3 276  494.41  423.51   27.0  2054.0  2027.0  25.49
pc.new.stud.from.top.10pc    4 276   18.51    9.87    1.0    56.0    55.0   0.59
pc.new.stud.from.top.25pc    5 276   44.89   15.35    9.0    87.0    78.0   0.92
num.ft.undergrad             6 276 2202.79 2153.70  249.0 11493.0 11244.0 129.64
num.pt.undergrad             7 276  624.53  766.06    6.0  5346.0  5340.0  46.11
in.state.tuition             8 276 7205.42 3858.08  608.0 15476.0 14868.0 232.23
out.of.state.tuition         9 276 8312.97 2657.17 1044.0 15476.0 14432.0 159.94
room                        10 276 1992.87  589.83  640.0  4358.0  3718.0  35.50
board                       11 276 1922.36  506.56  531.0  3700.0  3169.0  30.49
add.fees                    12 276  360.64  338.55   10.0  2147.0  2137.0  20.38
estim.book.costs            13 276  539.87  172.01   90.0  2340.0  2250.0  10.35
estim.personal.             14 276 1384.72  693.24  250.0  6800.0  6550.0  41.73
pc.fac.w.phd                15 276   64.33   15.37    8.0   103.0    95.0   0.93
stud..fac.ratio             16 276   14.95    3.59    4.6    28.8    24.2   0.22
graduation.rate             17 276   57.92   16.10   15.0   118.0   103.0   0.97
cluster*                    18 276    2.00    0.00    2.0     2.0     0.0   0.00
```

```
------------------------------------------------------------------------------------------------------------
group: 3
                          vars   n     mean      sd    min     max   range     se
num.appli.rec.d              1 149  3338.74 2982.25  212.0 13865.0 13653.0 244.32
num.appl.accepted            2 149  1930.68 1402.80  189.0  7260.0  7071.0 114.92
num.new.stud.enrolled        3 149   619.95  432.66   91.0  2464.0  2373.0  35.44
pc.new.stud.from.top.10pc    4 149    44.85   19.41   16.0    96.0    80.0   1.59
pc.new.stud.from.top.25pc    5 149    73.99   14.87   40.0   100.0    60.0   1.22
num.ft.undergrad             6 149  2445.71 1818.16  309.0 10142.0  9833.0 148.95
num.pt.undergrad             7 149   283.26  505.68    1.0  4379.0  4378.0  41.43
in.state.tuition             8 149 15272.43 3283.30 2650.0 20100.0 17450.0 268.98
out.of.state.tuition         9 149 15420.20 2946.58 6550.0 20100.0 13550.0 241.39
room                        10 149  2652.44  725.69 1025.0  4816.0  3791.0  59.45
board                       11 149  2522.19  491.77 1550.0  4541.0  2991.0  40.29
add.fees                    12 149   358.65  310.63   36.0  1836.0  1800.0  25.45
estim.book.costs            13 149   557.05  157.60  300.0  1495.0  1195.0  12.91
estim.personal.             14 149   979.55  407.79  300.0  3160.0  2860.0  33.41
pc.fac.w.phd                15 149    86.13    8.70   56.0   100.0    44.0   0.71
stud..fac.ratio             16 149    11.40    2.97    2.9    21.5    18.6   0.24
graduation.rate             17 149    81.14   11.83   42.0   100.0    58.0   0.97
cluster*                    18 149     3.00    0.00    3.0     3.0     0.0   0.00
```

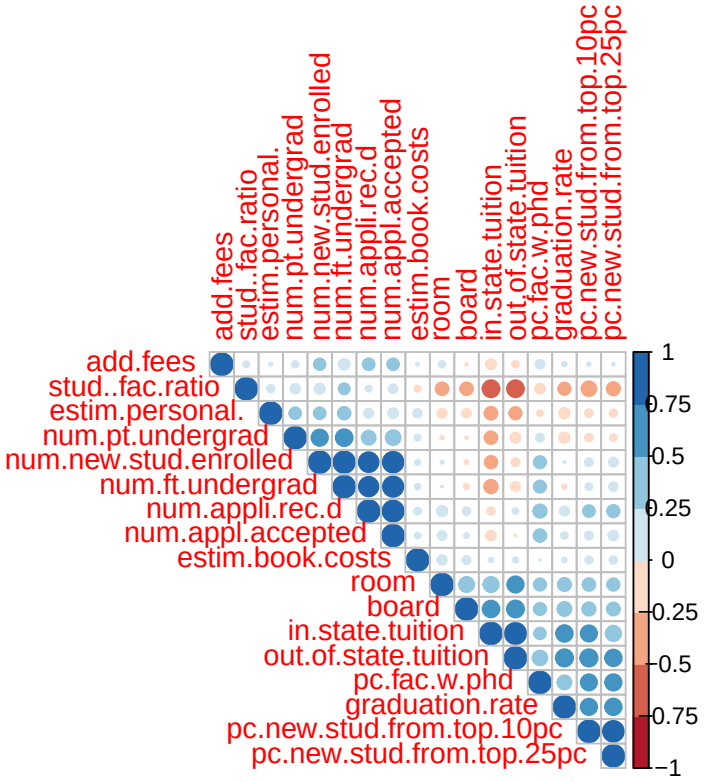
DescribeBy function in R allows us the get important statistics information from our groups. Some insights are the following:

- **Group 1**: Remember that the standard deviation measures the variability, and here we can see how this group has a lot of variability in the number of applications received compare to the other groups. By seen the median, min, max, and range we can determine that this group receives a lot of applications compared the cluster 2 and 3. Additionally, this group has a huge difference in tuition fees for its in-state and out-of-state students, and by seen its min values the minimum tuition fee of an in-state student is $672, whereas for an out-of-state student the minimum fee is $4,104.
- **Group 2**: Compare cluster 1 and 3, this group have similar tuition fees for its in-state and out-of-state students. Further, this group has the lowest graduation rate and the lowest percentage of new students in the top 10% out of the three clusters.
- **Group 3**: This group has the highest graduation rate, and something interesting is that it has the lowest ratio between number of students and faculty.

Even though the describeBy functions allowed us the get some information from these groups, it is not the best way to interpret the data. Creating visuals is a much better approach to understand the data and identify the patterns.

Let's plot a correlation matrix that will help us to summarize data from multiple variables. This plot type might not be the best to get an advanced analysis, but it will help us to get a first look diagnosis and from here we could see which variables would be interesting to analyze further.

```r
# Multivariate correlation matrix
corrplot(cor(df_cluster_unnorm[,1:17]), type = "upper", order = "hclust",
         col = brewer.pal(n = 8, name = "RdBu"))
```



Remember that correlation does not mean causation. So, these plots will help us to analyze how to data is correlated and it might help us to get some insights. Here we see the following patterns:

- There is a positive correlation between:

  New students enrolled full-time.

  Out-of-state tuition and graduation rate.

  Room price and out-of-state tuition.

- There is a negative correlation between:

  Out-of-state tuition and the ratio between the number of students and faculty.

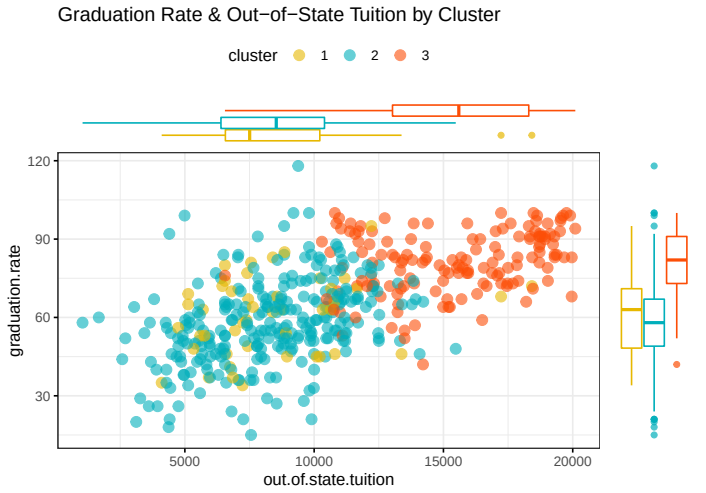
  In-state tuition and the number of full-time students.

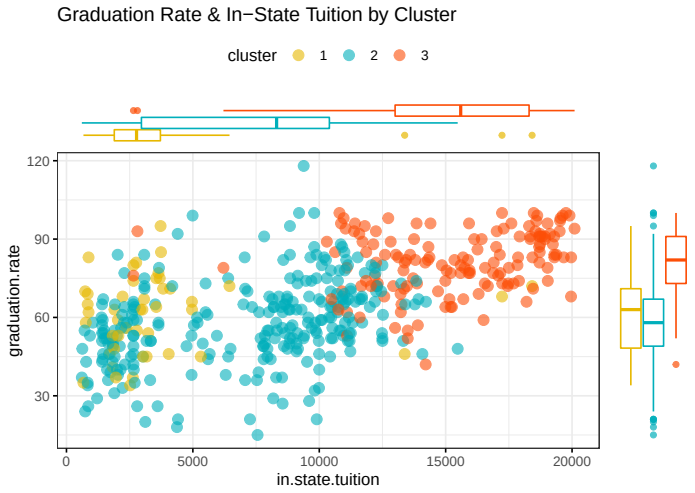
Now, let's create more detailed visuals.

```
# Scatter plot of graduation rate and out-of-state tuition & In-State fees by cluster
ggscatterhist( df_cluster_unnorm, x = "out.of.state.tuition", y = "graduation.rate",
  title = "Graduation Rate & Out-of-State Tuition by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())

ggscatterhist( df_cluster_unnorm, x = "in.state.tuition", y = "graduation.rate",
  title = "Graduation Rate & In-State Tuition by Cluster",
```

```
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())
```



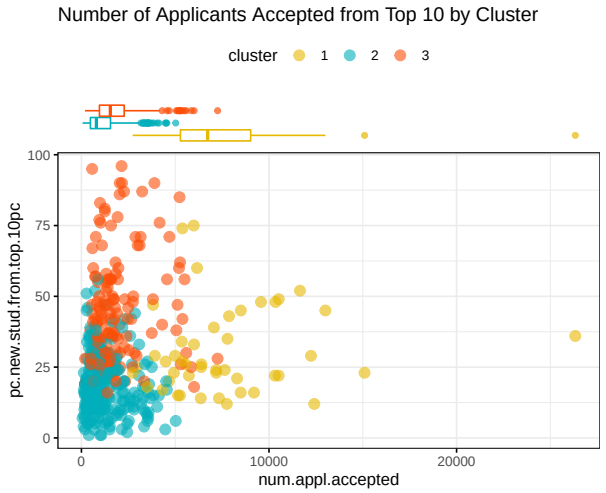


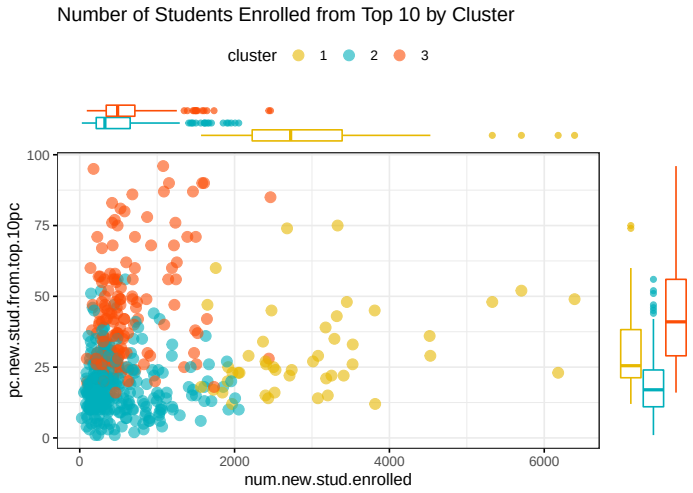
As we can see, cluster 3 has universities with higher graduation rates and also higher tuition fees in both out-of-state and in-state fees, in which 75% of its data are around 6800 and 18500.

Clusters 1 and 2 seem to have similar graduation rates and out-of-state fees. Regarding in-state tuition fees, we can see how cluster 1 charges much less tuition to their in-state students, and some universities from cluster 2 do the same.

```
# Scatter plot of number of applicants accepted and enrolled from students top 10 by cluster
ggscatterhist( df_cluster_unnorm, x = "num.appl.accepted", y = "pc.new.stud.from.top.10pc",
  title = "Number of Applicants Accepted from Top 10 by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())

ggscatterhist( df_cluster_unnorm, x = "num.new.stud.enrolled",
  y = "pc.new.stud.from.top.10pc",
  title = "Number of Students Enrolled from Top 10 by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())
```





Here we can see that the clusters behaves similarly based on the number of applicants accepted and students enrolled in their universities. Something interesting is that cluster 1 accepts more applicants, but the number of students that enroll in a program is fewer, which shows a lot of spread on its data.

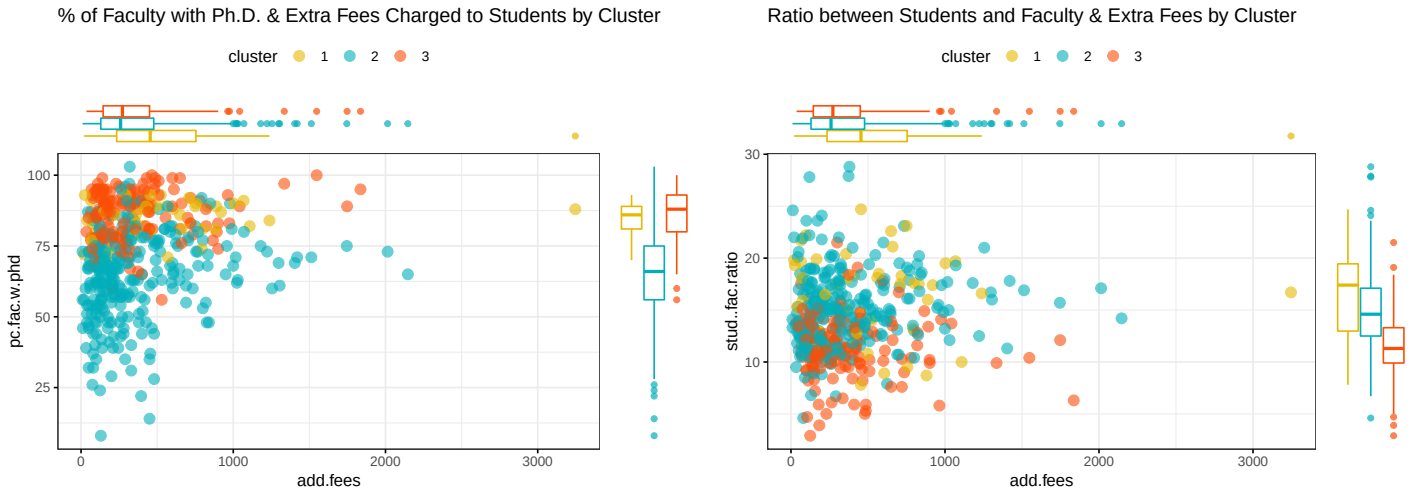
```
# Scatter plot of the graduation rate and the personal expenses by cluster
ggscatterhist( df_cluster_unnorm, x = "estim.personal.", y = "graduation.rate",
  title = "Graduation Rate & Estimation of Personal Expenses by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())
```

This chart shows the estimation of personal expenses and their relationship with the graduation rate. Something very interesting is that cluster 3 has the highest graduation rate and a lowest estimation of personal expenses.

On the other hand, clusters 1 and 2 have similar graduation rates, but cluster 1 has higher personal expenses than cluster 2 with 75% of its data between 700 and 2400.

```
# Scatter plot showing the ratio between students-faculty, %faculty with Ph.D. & the extra fees charged
ggscatterhist( df_cluster_unnorm, x = "add.fees", y = "pc.fac.w.phd",
  title = "% of Faculty with Ph.D. & Extra Fees Charged to Students by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())

ggscatterhist( df_cluster_unnorm, x = "add.fees", y = "stud..fac.ratio",
  title = "Ratio between Students and Faculty & Extra Fees by Cluster",
  color = "cluster", size = 3, alpha = 0.6,
  palette = c("#E7B800", "#00AFBB", "#FC4E07"),
  margin.plot = "boxplot",
  ggtheme = theme_bw())
```



These scatterplots are very interesting to analyze. It compares the additional fees charged based on the percentage of faculty with a doctorate and the ratio between the number of students and faculty.

As the graphs show, cluster 3 has the highest percentage of faculty with a Ph.D. and the lowest ratio between students and faculty. Followed by cluster 1, with a highly 50% concentrated data between 80 to 85% of faculty with Ph.D., and its additional fees tend to be higher than the other clusters.

4. Use the categorical measurements that were not used in the analysis (State and Private/Public) to characterize the different clusters. Is there any relationship between the clusters and the categorical information?

Create a new data frame combining numerical and categotical data, and also adding the cluster column.

```
# Create the data frame
df_categ_cluster <- data.frame(na.omit(Universities), cluster = k3$cluster)

# See the number of data points on each cluster by private and public institution
cluster1 <- df_categ_cluster %>% group_by(public..1...private..2.) %>%
                                 filter(cluster == 1) %>%
                                 summarise(Total = n())

cluster2 <- df_categ_cluster %>% group_by(public..1...private..2.) %>%
                                 filter(cluster == 2) %>%
```

```
                                  summarise(Total = n())

cluster3 <- df_categ_cluster %>% group_by(public..1...private..2.) %>%
                                  filter(cluster == 3) %>%
                                  summarise(Total = n())

# To see the values
as.data.frame(cluster1)
```

```
  public..1...private..2. Total
1                       1    41
2                       2     5
```

```
as.data.frame(cluster2)
```

```
  public..1...private..2. Total
1                       1    83
2                       2   193
```

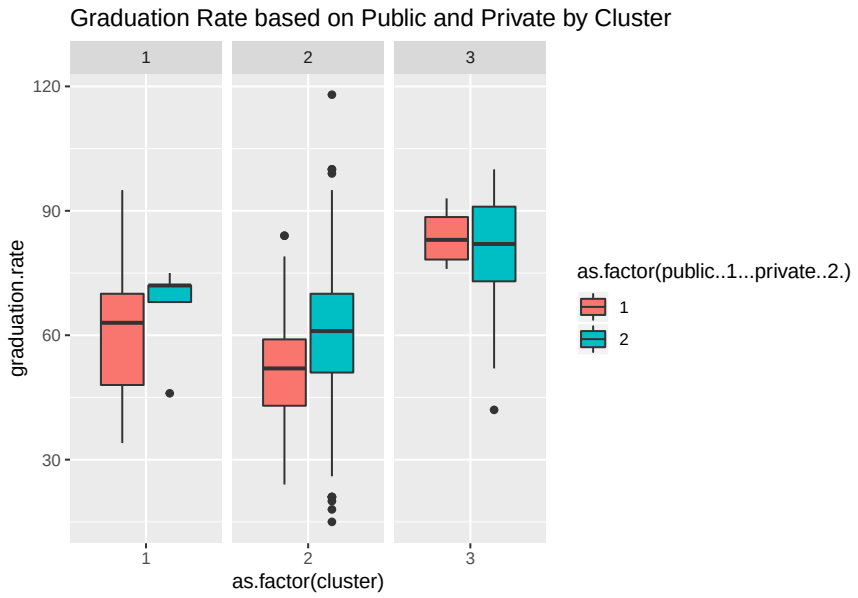
```
as.data.frame(cluster3)
```

```
  public..1...private..2. Total
1                       1     4
2                       2   145
```

These tables show that the biggest cluster is cluster 2 wit a total of 276. Cluster 2 and 3 have more private universities, and cluster 1 has more public universities.

Now, let's do some visualizations.

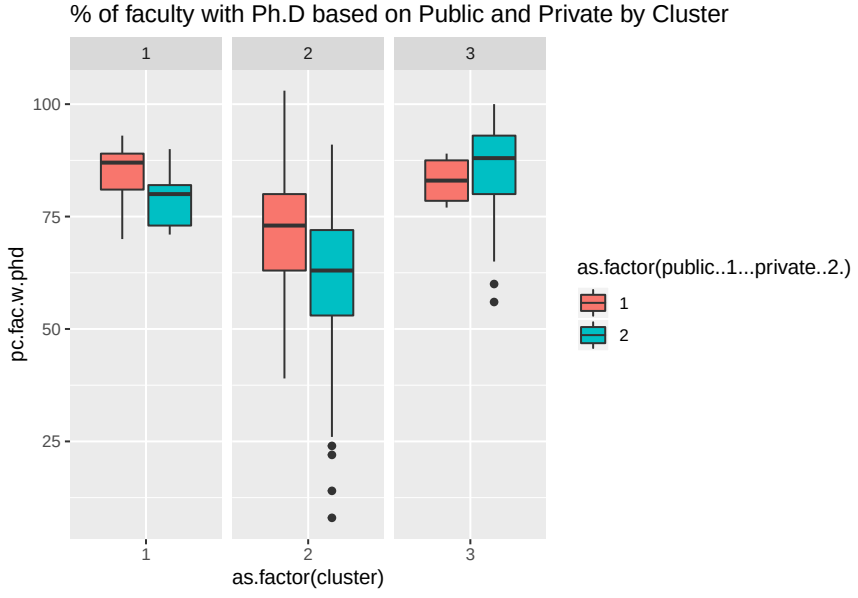
```
# Box plot of graduation rate on each cluster as public and private
ggplot(df_categ_cluster, aes(x=as.factor(cluster), y=graduation.rate,
                             fill=as.factor(public..1...private..2.))) + geom_boxplot() +
                         facet_wrap(~as.factor(cluster), scale="free_x") +
                         ggtitle("Graduation Rate based on Public and Private by Cluster")
```



As we saw earlier, cluster 3 has the highest graduation rate. Private universities have a higher spread compare to the public, but it might be because this cluster has 4 public universities and 145 private universities.

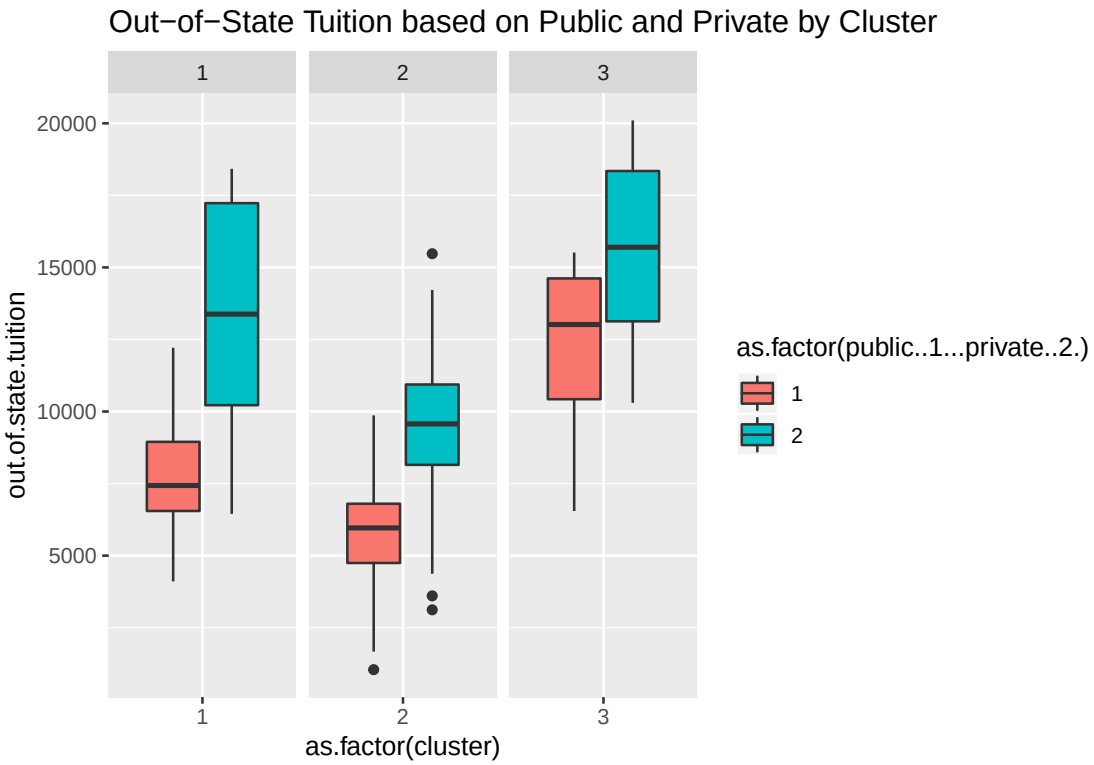
Cluster 2 has a lot of outliers, and cluster 1 has a big range of graduation range between its public universities. However, private universities in cluster 1 have higer graduation rate than public colleges.

```
# Box plot of percentage of faculty with Ph.D on each cluster as public and private
ggplot(df_categ_cluster, aes(x=as.factor(cluster), y=pc.fac.w.phd,
                             fill=as.factor(public..1...private..2.))) + geom_boxplot()  +
                        facet_wrap(~as.factor(cluster), scale="free_x") +
                        ggtitle("% of faculty with Ph.D based on Public and Private by Cluster")
```



In the graph shown above, the highest percentage of faculty with Ph.D. is cluster 3 with private universities. And, cluster 1 has a bit lower percent in public universities. Regarding cluster 2, we can see that 50% of its data is wider comapre the other two clusters.

```r
# Box plot of out-of-state tuition based on public and private
ggplot(df_categ_cluster, aes(x=as.factor(cluster), y=out.of.state.tuition,
                             fill=as.factor(public..1...private..2.))) + geom_boxplot() +
                          facet_wrap(~as.factor(cluster), scale="free_x") +
                          ggtitle("Out-of-State Tuition based on Public and Private by Cluster")
```



Here we can see a relationship between the clusters and the university type, which shows that private universities charge higher tuition fees to out-of-state students on every cluster. Analyzing between clusters, this chart shows that custer 2 charges less tuition compare to cluster 1 and 3.

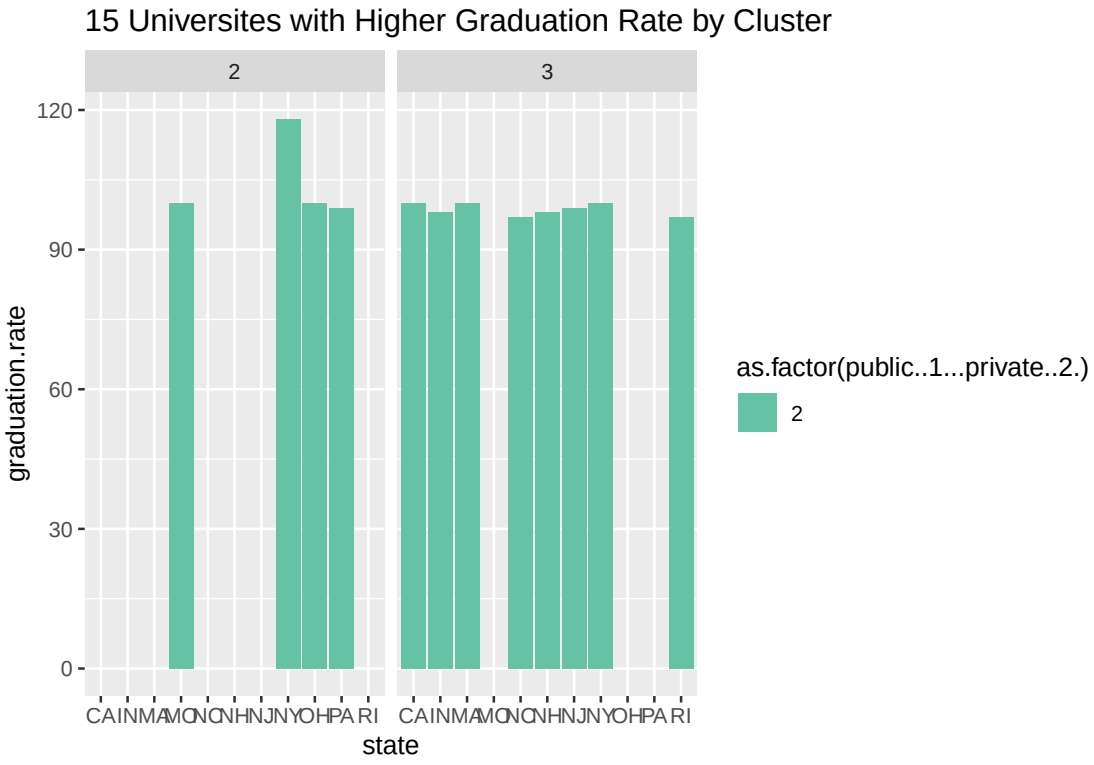
Now, to analyze the data from states, let's create a new data frame and take the top 15 states with the higher and lower graduation rate.

```r
# Higer and lower graduation rate
highest_15 <- df_categ_cluster %>% select(c(1:21)) %>%
                            setorder(graduation.rate) %>%
                            top_n(15, graduation.rate)

lowest_15 <- df_categ_cluster %>% select(c(1:21)) %>%
                            setorder(graduation.rate) %>%
                            top_n(-15, graduation.rate)
```

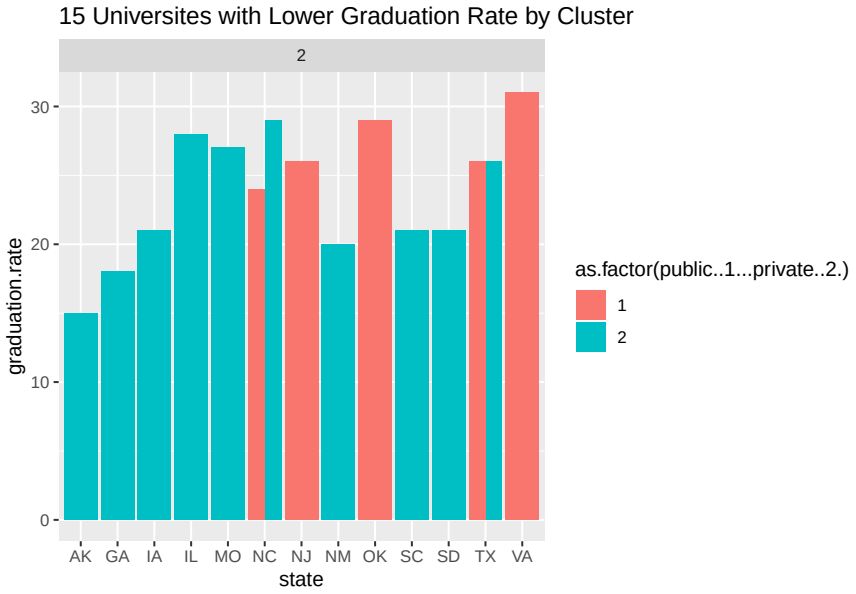
```r
# highest_15
ggplot(highest_15, aes(x = state, y = graduation.rate)) +
  geom_bar(aes(fill = as.factor(public..1...private..2.)),
           stat = "identity", #color = "white",
           position = position_dodge(0.9)) +
  facet_wrap(~as.factor(cluster)) +
```

```
  fill_palette("Set2") +
  ggtitle("15 Universites with Higher Graduation Rate by Cluster")
```



As we can see in this bar plot, the top 15 with higher graduation rate are from clusters 2 and 3, and something surprising is that all universities are private. The highest graduation rate is a university from New York on cluster 2.

```
# lowest_15
ggplot(lowest_15, aes(x = state, y = graduation.rate)) +
  geom_bar(aes(fill = as.factor(public..1...private..2.)),
           stat = "identity", #color = "white",
           position = position_dodge(0.9)) +
  facet_wrap(~as.factor(cluster)) +
  fill_palette("Acceder") +
  ggtitle("15 Universites with Lower Graduation Rate by Cluster")
```



The lowest 15 universities with very low graduation rate are from cluster 2, which is surprising because the university with highest graduation rate is also from cluster 2. This explains the outliers we got in previous charts.

Five universities are public and ten universities private. This visual also shows that the private university in the state of North Carolina has lower graduation rate than the public; and the state with lowest graduation rate is from Alaska state.

5. What other external information can explain the contents of some or all of these clusters?

Some external information that could help explain the clusters are the rate of international student accepted, ethnicity, average family income, number of financial aid and/or scholarships available to students.

6. Consider Tufts University, which is missing some information. Compute the Euclidean distance of this record from each of the clusters that you found above (using only the measurements that you have). Which cluster is it closest to? Impute the missing values for Tufts by taking the average of the cluster on those measurements.

Before calculating the distance of Tufts University's data to determine which group it will belong, we need to perform the following steps.

- First, we will create a new data frame with the new obervation, and we will drop the categorical values.
- It is essential to normalize the values to calculate the distance. To perform it, we will create a data frame by clusters, compute the mean and standard deviation, then apply it to our new obervation (Tufts University) to normalize it. In other words, we will use the z-score to normalize the data.

$$Z = \frac{X - \bar{X}}{SD}$$

Or formally written as:

$$Z = \frac{x - \mu}{\sigma}$$

- We will get the centroids, combine it with Tufts University data frame, and then we will calculate calculate the distance to determine to which cluster the new obervation will belong.
- Finally, impute the missing values by taking the closest cluster to the Tufts University data frame.

Let's start!

```
# tufts variable
df_tufts <- Universities  %>% filter(college.name == 'Tufts University')

# Drop categorical values before normalize our new data point
df_tufts <- df_tufts[, -c(1:3)]

# Select the dataframe by cluster to get the normalized values
df_cluster1 <- df_categ_cluster %>% filter(cluster == 1)
df_cluster2 <- df_categ_cluster %>% filter(cluster == 2)
df_cluster3 <- df_categ_cluster %>% filter(cluster == 3)

# To NORMALIZE data, we can use preProcess and scale function, but I was getting an error. So
# I will normalize for each cluster manually.

# To normalize cluster 1
avg1 = sapply(df_cluster1[, 4:20],function(x){return(mean(x,2))})
sd1 = sapply(df_cluster1[, 4:20],function(x){return(var(x))})
df_tufts1 = (df_tufts-avg1)/sd1

# To show the value
as.data.frame(df_tufts1)
```

```
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1   -4.370508e-05     -0.0001960983          -0.001138392
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                  0.148175                 0.1119183    -0.0002951732
  num.pt.undergrad in-state.tuition out-of-state.tuition        room
1               NA      0.001252817          0.001393068 0.002244794
        board     add.fees estim.book.costs estim.personal. pc.fac.w/phd
1 0.004652518 0.0001798749     0.0007044398    -0.001576658    0.3348681
  stud./fac.ratio graduation.rate
1      -0.4056355       0.1367077
```

```
# To normalize cluster 2
avg2 = sapply(df_cluster2[, 4:20],function(x){return(mean(x,2))})
sd2 = sapply(df_cluster2[, 4:20],function(x){return(var(x))})
df_tufts2 = (df_tufts-avg2)/sd2

# To show the value
as.data.frame(df_tufts2)
```

```
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1      0.00238364       0.002594516           0.004906351
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                 0.4413801                 0.1952115     0.0007114507
  num.pt.undergrad in-state.tuition out-of-state.tuition        room
```

```
1               NA     0.0007651095          0.001581465 0.003170509
        board   add.fees estim.book.costs estim.personal. pc.fac.w/phd
1 0.003955454 0.00212014      0.003380006   -0.0007532468    0.1397207
  stud./fac.ratio graduation.rate
1      -0.3344998       0.1311761
```

```
# To normalize cluster 3
avg3 = sapply(df_cluster3[, 4:20],function(x){return(mean(x,2))})
sd3 = sapply(df_cluster3[, 4:20],function(x){return(var(x))})
df_tufts3 = (df_tufts-avg3)/sd3

# To show the value
as.data.frame(df_tufts3)
```

```
  num.appli.rec'd num.appl.accepted num.new.stud.enrolled
1     0.000611661       0.001049363            0.00382491
  pc.new.stud.from.top.10pc pc.new.stud.from.top.25pc num.ft.undergrad
1                0.05041527                0.07232523      0.000835827
  num.pt.undergrad in-state.tuition out-of-state.tuition        room
1               NA     0.0003808877         0.0004729137 0.000793725
        board    add.fees estim.book.costs estim.personal. pc.fac.w/phd
1 0.002191565 0.002393997      0.004026015    0.0001683767    0.1452104
  stud./fac.ratio graduation.rate
1      -0.1131007      0.07151126
```

```
# Get the centroids of each cluster
centroid_1 <- k3$centers[1, ]
centroid_2 <- k3$centers[2, ]
centroid_3 <- k3$centers[3, ]

# Combine the centroid data with Tufts University's normalized data
tufts_centroid_1 <- bind_rows(centroid_1, df_tufts1)
tufts_centroid_2 <- bind_rows(centroid_2, df_tufts2)
tufts_centroid_3 <- bind_rows(centroid_3, df_tufts3)

# Compute the distances. Remember Euclidean distance is used by default.
tufts_distance1 <- get_dist(tufts_centroid_1)
tufts_distance2 <- get_dist(tufts_centroid_2)
tufts_distance3 <- get_dist(tufts_centroid_3)

# To show the results
tufts_distance1
```

```
         1
2 5.428572
```

```
tufts_distance2
```

```
         1
2 1.477714
```

```
tufts_distance3
```

```
         1
2 3.104426
```

This output shows that the closest cluster to Tufts University data point is cluster 2, with a Euclidean distance of 1.477714.

Now, let's impute the number of part-time undergrad students missing value.

```
# See my data frame with missing value
df_tufts2[, 5:7]
```

```
  pc.new.stud.from.top.25pc num.ft.undergrad num.pt.undergrad
1                 0.1952115     0.0007114507               NA
```

```
# Impute/replace the missing value from centroid 2
df_tufts2$num.pt.undergrad <- as.vector(centroid_2)[7]

# See my data frame with imputed value
df_tufts2[, 5:7]
```

```
  pc.new.stud.from.top.25pc num.ft.undergrad num.pt.undergrad
1                 0.1952115     0.0007114507       -0.2732909
```

Finally, we can see how the number of part-time undergrad students have been fill out fron the centroid value from cluster 2.