clustering exercise

Julia Sheriff

10/27/2018

This mini-project is based on the K-Means exercise from 'R in Action' Go here for the original blog post and solutions http://www.r-bloggers.com/k-means-clustering-from-r-in-action/

Exercise 0: Install these packages if you don't have them already

install.packages("cluster") install.packages("rattle.data") install.packages("NbClust")

load the data and look at the first few rows

```
data(wine, package="rattle.data")
head(wine)
```

```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39
##   Nonflavanoids Proanthocyanins Color  Hue Dilution Proline
## 1          0.28            2.29  5.64 1.04     3.92    1065
## 2          0.26            1.28  4.38 1.05     3.40    1050
## 3          0.30            2.81  5.68 1.03     3.17    1185
## 4          0.24            2.18  7.80 0.86     3.45    1480
## 5          0.39            1.82  4.32 1.04     2.93     735
## 6          0.34            1.97  6.75 1.05     2.85    1450
```

```
str(wine)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Type           : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol        : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash            : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity     : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium      : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Phenols        : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids     : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavanoids  : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color          : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue            : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ Dilution       : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline        : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

HELP- Whats the difference between the code below versus:

data2 <- wine[, 2:14]?

Is the code below the equivalent of developing a kmeans training set?

```
data <- scale(wine[-1])
str(data)
```

```
##  num [1:178, 1:13] 1.514 0.246 0.196 1.687 0.295 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
##  - attr(*, "scaled:center")= Named num [1:13] 13 2.34 2.37 19.49 99.74 ...
##   ..- attr(*, "names")= chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
##  - attr(*, "scaled:scale")= Named num [1:13] 0.812 1.117 0.274 3.34 14.282 ...
##   ..- attr(*, "names")= chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
```

HELP:

Talk me through the code

How do I interpret these two graphs? What are the most important considerations?

Method 1-deciding number of clusters:

- A plot of the total within-groups sums of squares against the number of clusters in a K-means solution can be helpful.
- A bend in the graph can suggest the appropriate number of clusters.

---

length(data) wssplot <- function(data, nc=15, seed=1234){ wss <- (nrow(data)-1)*sum(apply(data,2,var)) for (i in 2:nc){ set.seed(seed) wss[i] <- sum(kmeans(data, centers=i)$withinss)}

```
        plot(1:nc, wss, type="b", xlab="Number of Clusters",
                     ylab="Within groups sum of squares")
}
```

wssplot(df)

- How many clusters does this method suggest?
- ???
- Why does this method work? What's the intuition behind it? We want clusters to be meaningful.
- A good number of clusters will clearly reduce the sum of squares, when comparing to an (n-1) number of clusters

---

Method 2 for figuring out # of clusters:

Use the NbClust library, which runs many experiments and gives a distribution of potential number of clusters.

---

library(NbClust) set.seed(1234) nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans") View(nc) nctable <- table(nc$Best.n[1,])$View(nctable)$barplot(table(nc$Best.n[1,]), xlab="Numer of Clusters", ylab="Number of Criteria", main="Number of Clusters Chosen by 26 Criteria") —

How many clusters does this method suggest? * ?? * ??

Exercise 4: Once you've picked the number of clusters, run k-means

HELP:

What's most important to consider in the output of str()?

Output the result of calling kmeans() into a variable fit.km * adding nstart=25 will generate 25 initial configurations. This approach is often recommended.

```
#exercise suggested 3
fit.km3 <- kmeans(data, 3, nstart = 25)
str(fit.km3)
```

```
## List of 9
##  $ cluster     : int [1:178] 2 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:3, 1:13] 0.164 0.833 -0.923 0.869 -0.303 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
##  $ totss       : num 2301
##  $ withinss    : num [1:3] 326 386 559
##  $ tot.withinss: num 1271
##  $ betweenss   : num 1030
##  $ size        : int [1:3] 51 62 65
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
fit.km3$size
```

```
## [1] 51 62 65
```

```
fit.km3$centers
```

```
##      Alcohol      Malic        Ash Alcalinity   Magnesium     Phenols
## 1  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
## 2  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
## 3 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
##    Flavanoids Nonflavanoids Proanthocyanins      Color        Hue
## 1 -1.21182921    0.72402116     -0.77751312  0.9388902 -1.1615122
## 2  0.97506900   -0.56050853      0.57865427  0.1705823  0.4726504
## 3  0.02075402   -0.03343924      0.05810161 -0.8993770  0.4605046
```

```
##     Dilution    Proline
## 1 -1.2887761 -0.4059428
## 2  0.7770551  1.1220202
## 3  0.2700025 -0.7517257
```

```
aggregate(wine[-1], by = list(cluster = fit.km3$cluster), mean)
```

```
##   cluster  Alcohol    Malic      Ash Alcalinity Magnesium  Phenols
## 1       1 13.13412 3.307255 2.417647   21.24118  98.66667 1.683922
## 2       2 13.67677 1.997903 2.466290   17.46290 107.96774 2.847581
## 3       3 12.25092 1.897385 2.231231   20.06308  92.73846 2.247692
##   Flavanoids Nonflavanoids Proanthocyanins    Color       Hue Dilution
## 1  0.8188235     0.4519608        1.145882 7.234706 0.6919608 1.696667
## 2  3.0032258     0.2920968        1.922097 5.453548 1.0654839 3.163387
## 3  2.0500000     0.3576923        1.624154 2.973077 1.0627077 2.803385
##     Proline
## 1  619.0588
## 2 1100.2258
## 3  510.1692
```

I want to try 4 also:

```
fit.km4 <- kmeans(data, 4, nstart = 25)
str(fit.km4)
```

```
## List of 9
##  $ cluster     : int [1:178] 2 2 2 2 3 2 2 2 2 2 ...
##  $ centers     : num [1:4, 1:13] 0.186 0.958 -0.787 -0.905 0.902 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
##  $ totss       : num 2301
##  $ withinss    : num [1:4] 303 269 307 290
##  $ tot.withinss: num 1169
##  $ betweenss   : num 1132
##  $ size        : int [1:4] 49 56 28 45
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
fit.km4$size
```

```
## [1] 49 56 28 45
```

```
fit.km4$centers
```

```
##      Alcohol       Malic        Ash Alcalinity   Magnesium    Phenols
## 1  0.1860184  0.90242582  0.2485092  0.5820616 -0.05049296 -0.9857762
## 2  0.9580555 -0.37748461  0.1969019 -0.8214121  0.39943022  0.9000233
## 3 -0.7869073  0.04195151  0.2157781  0.3683284  0.43818899  0.6543578
## 4 -0.9051690 -0.53898599 -0.6498944  0.1592193 -0.71473842 -0.4537841
##   Flavanoids Nonflavanoids Proanthocyanins      Color        Hue
## 1 -1.2327174     0.7148253      -0.7474990  0.9857177 -1.1879477
## 2  0.9848901    -0.6204018       0.5575193  0.2423047  0.4799084
## 3  0.5746004    -0.5429201       0.8888549 -0.7346332  0.2830335
## 4 -0.2408779     0.3315072      -0.4329238 -0.9177666  0.5202140
##      Dilution    Proline
## 1 -1.29787850 -0.3789756
```

```
## 2  0.76926636  1.2184972
## 3  0.60628629 -0.5169332
## 4  0.07869143 -0.7820425
```

```
aggregate(wine[-1], by = list(cluster = fit.km4$cluster), mean)
```

```
##   cluster  Alcohol    Malic      Ash Alcalinity Magnesium  Phenols
## 1       1 13.15163 3.344490 2.434694   21.43878  99.02041 1.678163
## 2       2 13.77839 1.914643 2.420536   16.75179 105.44643 2.858393
## 3       3 12.36179 2.383214 2.425714   20.72500 106.00000 2.704643
## 4       4 12.26578 1.734222 2.188222   20.02667  89.53333 2.011111
##   Flavanoids Nonflavanoids Proanthocyanins    Color       Hue Dilution
## 1  0.7979592     0.4508163        1.163061 7.343265 0.6859184 1.690204
## 2  3.0130357     0.2846429        1.910000 5.619821 1.0671429 3.157857
## 3  2.6032143     0.2942857        2.099643 3.355000 1.0221429 3.042143
## 4  1.7886667     0.4031111        1.343111 2.930444 1.0763556 2.667556
##     Proline
## 1  627.5510
## 2 1130.6071
## 3  584.1071
## 4  500.6222
```

HELP:

What does randIndex really do?

1. Using the table() function, show how the clusters in fit.km$clusters

- ct stands for cross tabulation, this process of evaluation

2. Use randIndex to evaluate

- The adjusted Rand index provides a measure of the agreement between two partitions, adjusted for chance. It ranges from -1 (no agreement) to 1 (perfect agreement). Agreement between the wine varietal type and the cluster solution is 0.9.

```
library(flexclust)
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
#with 3
ct.km3 <- table(wine$Type, fit.km3$cluster)
ct.km3
```

```
##
##      1  2  3
##   1  0 59  0
##   2  3  3 65
##   3 48  0  0
```

```
randIndex(ct.km3)
```

```
##      ARI
## 0.897495
```

```
#0.897495
```

with 4:

```
ct.km4 <- table(wine$Type, fit.km4$cluster)
ct.km4
```

```
##
##      1  2  3  4
##   1  0 55  4  0
##   2  1  1 24 45
##   3 48  0  0  0
```

```
randIndex(ct.km4)
```

```
##       ARI
## 0.7535909
```

```
#0.7535909
#using 3 clusters is better
```

HELP

Visualize these clusters using function clusplot() from the cluster library

- Would you consider this a good clustering?

library(cluster) clusplot() clusplot()