MA_Quiz2

Nilay Kamar

4/8/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

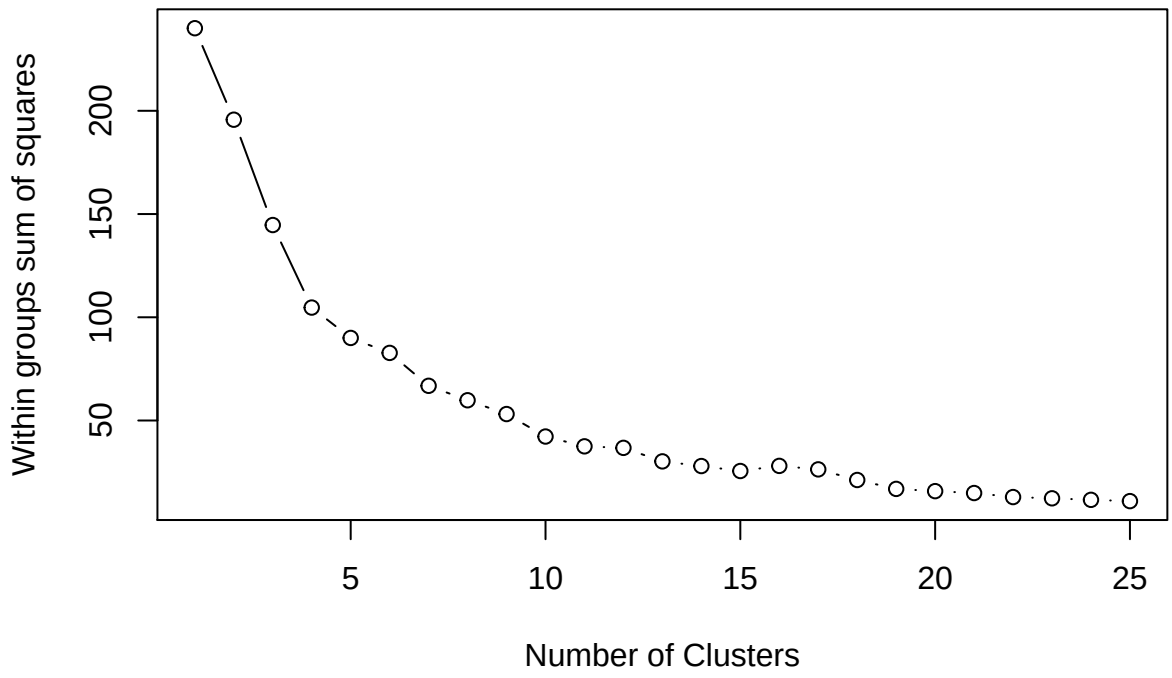
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(cluster)
library(purrr)


# Install and load packages
#install.packages("NbClust")
library(NbClust)
library(ggplot2)

# Load data with the name cldata
data <- read.csv("/Users/nilaykamar/Desktop/Marketing Analytics/w2/courseContent/Quiz 2 - Data US Cities

# Review data
summary(data)
```

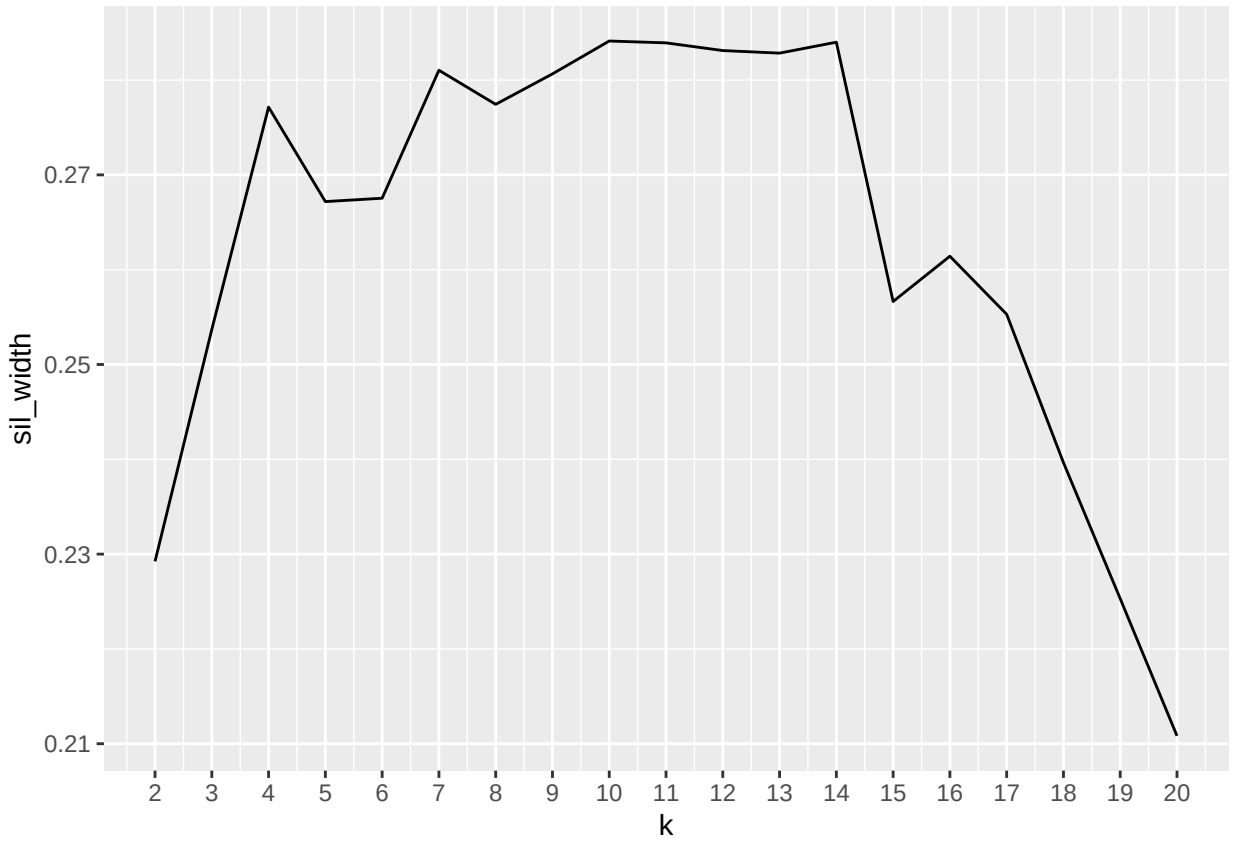
```
##           City      PercBlack        PercHisp       PercAsian
##  Albuquerque: 1   Min.   : 1.00   Min.   : 1.00   Min.   : 1.000
##  Atlanta    : 1   1st Qu.:11.00   1st Qu.: 3.00   1st Qu.: 1.000
##  Austin     : 1   Median :22.00   Median : 6.00   Median : 2.000
##  Baltimore  : 1   Mean   :24.35   Mean   :14.59   Mean   : 6.041
##  Boston     : 1   3rd Qu.:31.00   3rd Qu.:23.00   3rd Qu.: 5.000
##  Charlotte  : 1   Max.   :76.00   Max.   :69.00   Max.   :71.000
##  (Other)    :43
##    MedianAge      Unemployment
##  Min.   :28.00   Min.   : 3.00
##  1st Qu.:30.00   1st Qu.: 5.00
##  Median :32.00   Median : 7.00
##  Mean   :31.88   Mean   : 7.02
##  3rd Qu.:33.00   3rd Qu.: 9.00
##  Max.   :37.00   Max.   :13.00
##
```

```
# Scale data
testdata <- data[,2:6]
testdata <- scale(testdata)

# Determine number of clusters. Option 1: visual rule
#ELBOW ANALYSIS
wss <- (nrow(testdata)-1)*sum(apply(testdata,2,var))
```

```
for (i in 2:25) wss[i] <- sum(kmeans(testdata,
                                     centers=i)$withinss)
plot(1:25, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
```

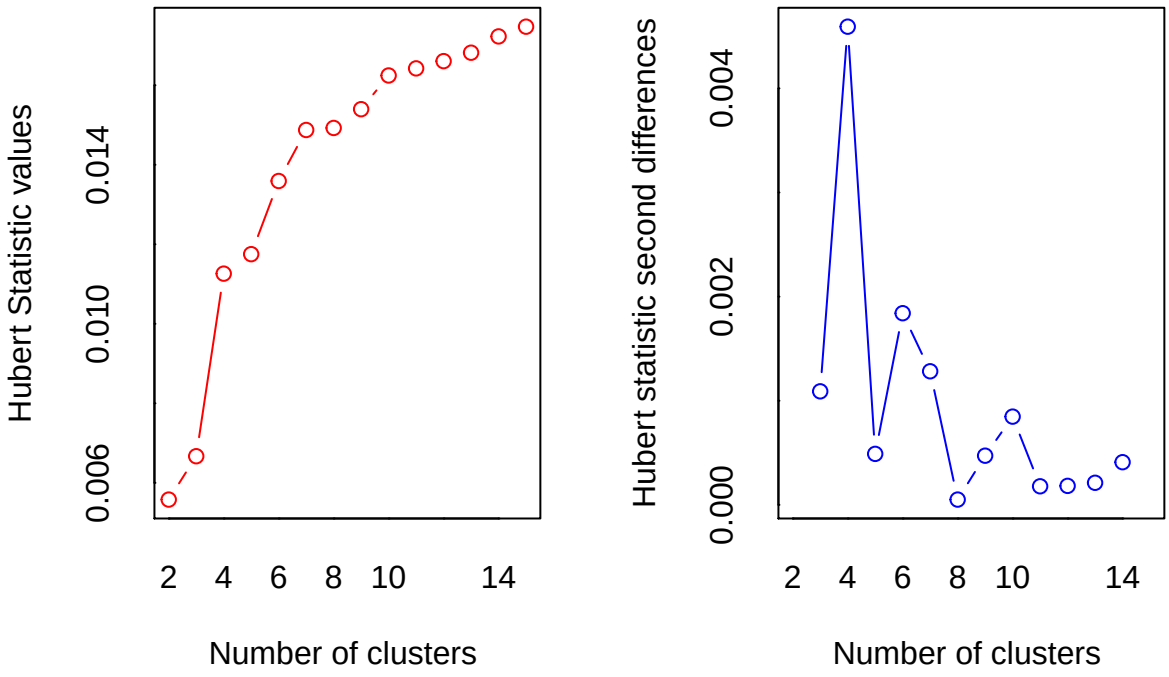



```
#Silhouette Analysis
sil_width <- map_dbl(2:20,  function(k){
  model <- pam(x = testdata, k = k)
  model$silinfo$avg.width
})

sil_df <- data.frame(
  k = 2:20,
  sil_width = sil_width
)

ggplot(sil_df, aes(x = k, y = sil_width)) +
  geom_line() +
  scale_x_continuous(breaks = 2:20)
```

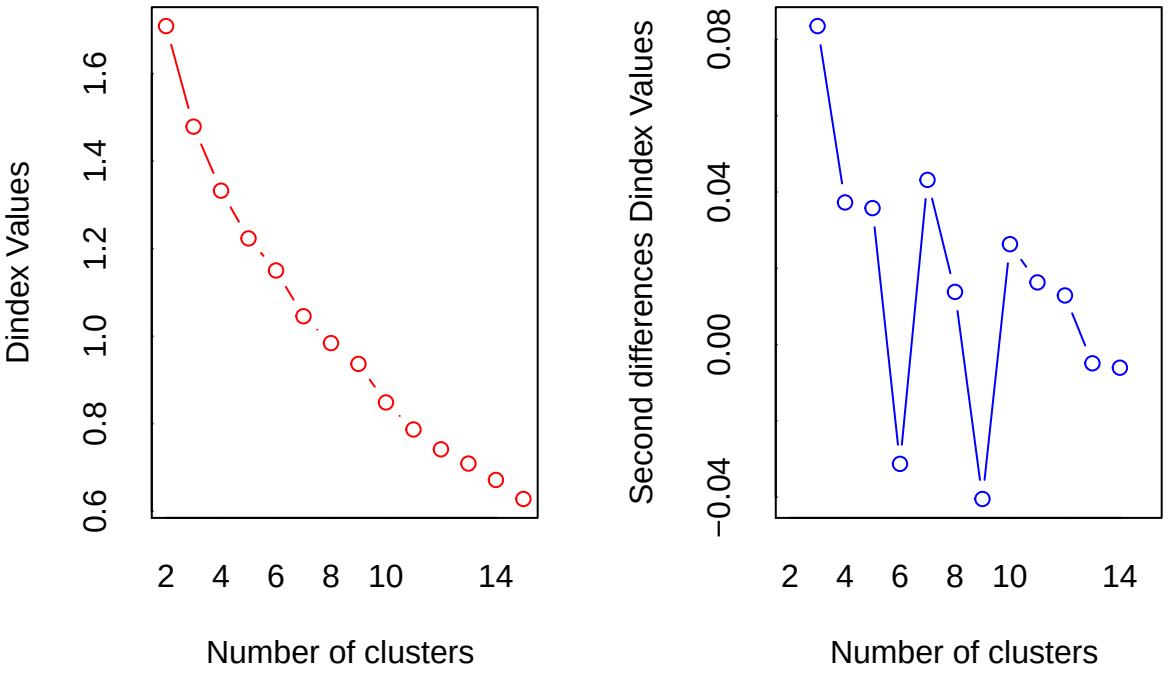



```
# Determine number of clusters. Option 2: more frequent optimal number
res <- NbClust(testdata, diss=NULL, distance = "euclidean",
     min.nc=2, max.nc=15,
     method = "kmeans", index = "all")
```

```
## Warning in pf(beale, pp, df2): NaNs produced
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
```

```
##                 the measure.
##
## *****************************************************************
## * Among all indices:
## * 4 proposed 2 as the best number of clusters
## * 3 proposed 3 as the best number of clusters
## * 7 proposed 4 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 6 proposed 15 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *****************************************************************
```

```
res$Best.partition
```

```
##  [1] 4 1 4 1 4 4 1 1 1 4 3 4 1 3 3 3 2 3 4 4 4 4 3 3 1 3 4 4 4 1 3 1 4 4 1
## [36] 4 4 4 3 1 3 3 2 3 4 4 4 4 4
```

```
# K-Means Cluster Analysis (based on the proposed number by NbCluster)
options(digits = 2)
fit <- kmeans(testdata, 4)
table(fit$cluster)
```

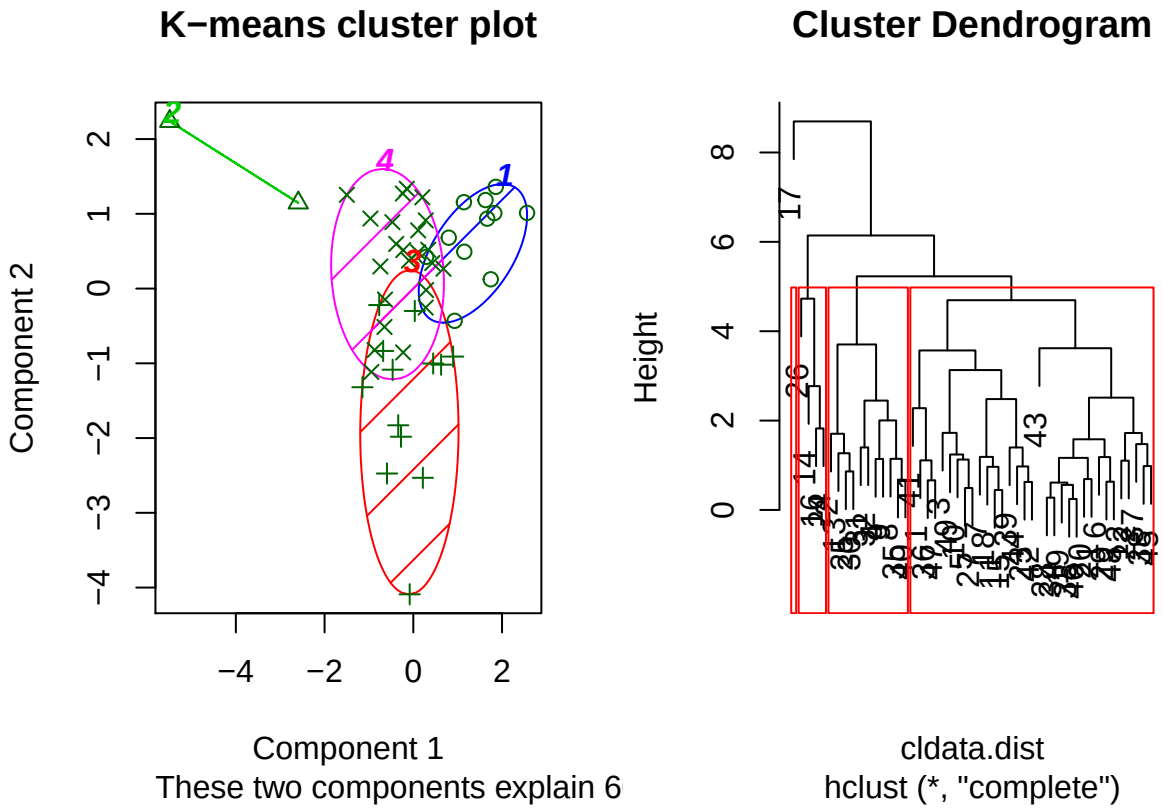
```
##
##  1  2  3  4
## 11  2 13 23
```

```
# Calculate average for each cluster
aggregate(data[,2:6],by=list(fit$cluster),FUN=mean)
```

```
##   Group.1 PercBlack PercHisp PercAsian MedianAge Unemployment
## 1       1        52      5.3       2.8        32          8.9
## 2       2         6      9.5      50.0        36          5.5
## 3       3        16     33.8       7.8        31          9.2
## 4       4        17      8.7       2.7        32          5.0
```

```
# Add segmentation to dataset
cldata.w.cluster <- data.frame(data, fit$cluster)


#EXTRA STEP
clusplot(testdata, fit$cluster, color=TRUE, shade=TRUE,
         labels=4, lines=0, main="K-means cluster plot")

#Hierarchical Clustering
```

```
cldata.dist <- dist(testdata)
cldata.hc <- hclust(cldata.dist, method="complete")

plot(cldata.hc)
rect.hclust(cldata.hc, k=4, border="red")
```




```
cldata.hc.segment <- cutree(cldata.hc, k=4)      # membership vector for 4 groups
table(cldata.hc.segment)
```

```
## cldata.hc.segment
##  1  2  3  4
## 33 11  4  1
```