Notebook 3 Clustering

Umaid Ahmed

Link to data set: Mushrooms.csv https://archive.ics.uci.edu/ml/datasets/Mushroom (https://archive.ics.uci.edu/ml/datasets/Mushroom)

Note: Some of the code chunks below are taken from Professor's github examples

Loading Data

lets load the data and see the structure of the dataset

```
mushrooms <- read.csv("mushrooms.csv", header = FALSE)
str(mushrooms)
```

```
## 'data.frame':    8125 obs. of  23 variables:
##  $ V1 : chr  "class" "p" "e" "e" ...
##  $ V2 : chr  "cap-shape" "x" "x" "b" ...
##  $ V3 : chr  "cap-surface" "s" "s" "s" ...
##  $ V4 : chr  "cap-color" "n" "y" "w" ...
##  $ V5 : chr  "bruises" "t" "t" "t" ...
##  $ V6 : chr  "odor" "p" "a" "l" ...
##  $ V7 : chr  "gill-attachment" "f" "f" "f" ...
##  $ V8 : chr  "gill-spacing" "c" "c" "c" ...
##  $ V9 : chr  "gill-size" "n" "b" "b" ...
##  $ V10: chr  "gill-color" "k" "k" "n" ...
##  $ V11: chr  "stalk-shape" "e" "e" "e" ...
##  $ V12: chr  "stalk-root" "e" "c" "c" ...
##  $ V13: chr  "stalk-surface-above-ring" "s" "s" "s" ...
##  $ V14: chr  "stalk-surface-below-ring" "s" "s" "s" ...
##  $ V15: chr  "stalk-color-above-ring" "w" "w" "w" ...
##  $ V16: chr  "stalk-color-below-ring" "w" "w" "w" ...
##  $ V17: chr  "veil-type" "p" "p" "p" ...
##  $ V18: chr  "veil-color" "w" "w" "w" ...
##  $ V19: chr  "ring-number" "o" "o" "o" ...
##  $ V20: chr  "ring-type" "p" "p" "p" ...
##  $ V21: chr  "spore-print-color" "k" "n" "n" ...
##  $ V22: chr  "population" "s" "n" "n" ...
##  $ V23: chr  "habitat" "u" "g" "m" ...
```

As it can be seen the data set contains char. We can turn features into factor and store the factors as integers

The function below is taken from my Group partner's notebook

```
mushrooms2 <- mushrooms
for(i in 1:ncol(mushrooms)){
  if(is.character(mushrooms[,i])){
    mushrooms[,i] <- as.factor(mushrooms[,i])
    mushrooms2[,i] <- as.factor(mushrooms[,i])
    mushrooms2[,i] <- as.integer(mushrooms[,i])
  }
}
str(mushrooms2)
```

```
## 'data.frame':    8125 obs. of  23 variables:
##  $ V1 : int  1 3 2 2 3 2 2 2 2 3 ...
##  $ V2 : int  3 7 7 1 7 7 7 1 1 7 ...
##  $ V3 : int  1 4 4 4 5 4 5 4 5 5 ...
##  $ V4 : int  3 6 11 10 10 5 11 10 10 10 ...
##  $ V5 : int  1 3 3 3 3 2 3 3 3 3 ...
##  $ V6 : int  7 8 1 4 8 6 1 1 4 8 ...
##  $ V7 : int  3 2 2 2 2 2 2 2 2 2 ...
##  $ V8 : int  2 1 1 1 1 3 1 1 1 1 ...
##  $ V9 : int  2 3 1 1 3 1 1 1 1 3 ...
##  $ V10: int  4 6 6 7 7 6 7 3 7 9 ...
##  $ V11: int  2 1 1 1 1 3 1 1 1 1 ...
##  $ V12: int  6 4 3 3 4 4 3 3 3 4 ...
##  $ V13: int  4 3 3 3 3 3 3 3 3 3 ...
##  $ V14: int  4 3 3 3 3 3 3 3 3 3 ...
##  $ V15: int  8 9 9 9 9 9 9 9 9 9 ...
##  $ V16: int  8 9 9 9 9 9 9 9 9 9 ...
##  $ V17: int  2 1 1 1 1 1 1 1 1 1 ...
##  $ V18: int  3 4 4 4 4 4 4 4 4 4 ...
##  $ V19: int  3 2 2 2 2 2 2 2 2 2 ...
##  $ V20: int  6 5 5 5 5 1 5 5 5 5 ...
##  $ V21: int  7 3 4 4 3 4 3 3 4 3 ...
##  $ V22: int  4 5 3 3 5 1 3 3 5 6 ...
##  $ V23: int  3 7 2 5 7 2 2 5 5 2 ...
```

dropping the class features of the dataset

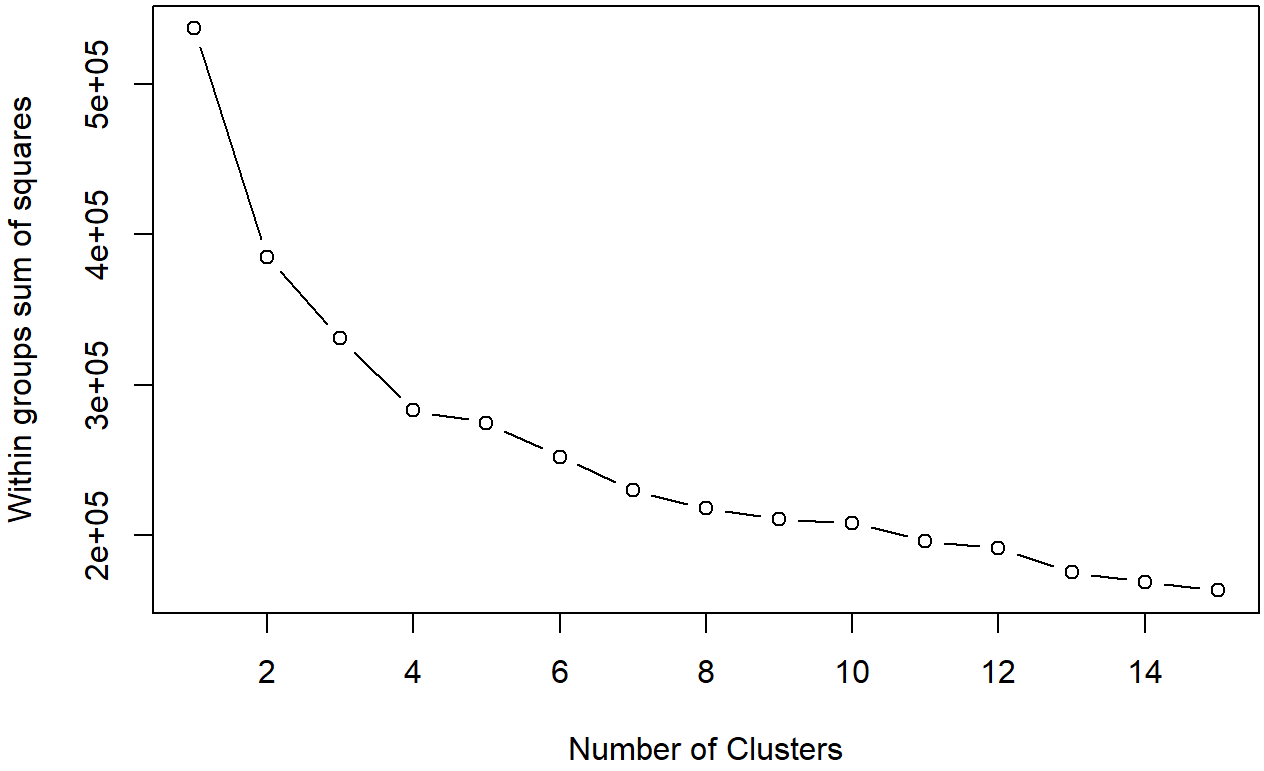
```
mushrooms3 <- mushrooms2[-1]
str(mushrooms3)
```

```
## 'data.frame':    8125 obs. of  22 variables:
##  $ V2 : int  3 7 7 1 7 7 7 1 1 7 ...
##  $ V3 : int  1 4 4 4 5 4 5 4 5 5 ...
##  $ V4 : int  3 6 11 10 10 5 11 10 10 10 ...
##  $ V5 : int  1 3 3 3 3 2 3 3 3 3 ...
##  $ V6 : int  7 8 1 4 8 6 1 1 4 8 ...
##  $ V7 : int  3 2 2 2 2 2 2 2 2 2 ...
##  $ V8 : int  2 1 1 1 1 3 1 1 1 1 ...
##  $ V9 : int  2 3 1 1 3 1 1 1 1 3 ...
##  $ V10: int  4 6 6 7 7 6 7 3 7 9 ...
##  $ V11: int  2 1 1 1 1 3 1 1 1 1 ...
##  $ V12: int  6 4 3 3 4 4 3 3 3 4 ...
##  $ V13: int  4 3 3 3 3 3 3 3 3 3 ...
##  $ V14: int  4 3 3 3 3 3 3 3 3 3 ...
##  $ V15: int  8 9 9 9 9 9 9 9 9 9 ...
##  $ V16: int  8 9 9 9 9 9 9 9 9 9 ...
##  $ V17: int  2 1 1 1 1 1 1 1 1 1 ...
##  $ V18: int  3 4 4 4 4 4 4 4 4 4 ...
##  $ V19: int  3 2 2 2 2 2 2 2 2 2 ...
##  $ V20: int  6 5 5 5 5 1 5 5 5 5 ...
##  $ V21: int  7 3 4 4 3 4 3 3 4 3 ...
##  $ V22: int  4 5 3 3 5 1 3 3 5 6 ...
##  $ V23: int  3 7 2 5 7 2 2 5 5 2 ...
```

KMEANS

In this section we would perform KMeans clustering.
First we would like to plot the within-groups sums of squares vs the number of clusters so that we can determine what are the best number of clusters for this data set. For this dataset we will plot clusters from 2 to 15 with seed 1234 for reproducibility.

```
wsplot <- 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")
}
wsplot(mushrooms3)
```



Here we would like to find if there is an elbow in the plot. WE can see a type of elbow forming on 4 clusters indicating that the best number of clusters for this dataset is 4.

Performing KMeans clustering with 4 clusters.

```
set.seed(1234)
KMean_clustering <- kmeans(mushrooms3, 4, nstart=20)
KMean_clustering
```

```
## K-means clustering with 4 clusters of sizes 1215, 2574, 1880, 2456
##
## Cluster means:
##         V2       V3       V4       V5       V6       V7       V8       V9
## 1 5.501235 3.498765 8.081481 2.000000 3.000000 2.000000 1.000000 1.000000
## 2 5.325175 3.654623 4.898601 2.806915 5.721834 1.918415 1.121212 1.113442
## 3 5.295745 4.461702 4.957447 2.055319 7.225532 2.000000 1.051064 2.838298
## 4 5.149837 3.685261 8.504479 2.486156 4.582248 2.000407 1.902687 1.519951
##        V10      V11      V12      V13      V14      V15      V16      V17
## 1 5.444444 1.000000 2.000000 2.000000 2.000000 4.183539 4.183539 1.000000
## 2 9.989510 2.452991 1.916472 2.876068 2.949106 6.917638 6.830614 1.000000
## 3 1.110638 2.838298 1.004255 2.527660 2.527660 7.927660 7.927660 1.000000
## 4 7.468241 1.822883 3.214577 2.590798 2.714577 8.983306 8.976792 1.000407
##       V18      V19      V20      V21      V22      V23
## 1 4.00000 2.000000 3.000000 2.000000 6.500412 3.041152
## 2 3.81352 2.184926 4.703963 4.226107 5.911033 1.914918
## 3 4.00000 2.161702 1.119149 8.987234 5.744681 3.844681
## 4 4.00285 2.156759 3.620928 3.921417 4.405130 2.996336
##
## Clustering vector:
##    [1] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##   [38] 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##   [75] 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [112] 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4
##  [149] 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [186] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [223] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4
##  [260] 4 4 2 4 4 4 4 4 4 4 4 2 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [297] 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4
##  [334] 4 4 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [371] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [408] 4 4 4 2 4 4 4 4 4 4 4 2 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [445] 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [482] 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [519] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [556] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4
##  [593] 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [630] 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4
##  [667] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 2 4 4 2 4 4 2 4
##  [704] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [741] 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [778] 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [815] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4
##  [852] 4 4 4 4 4 4 4 4 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [889] 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 2 4 4 4 4 4 4
##  [926] 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 2 4 4 4 4 2 4 4 4 4 4 4 4 2 4 4 4
##  [963] 4 2 2 4 4 2 2 4 4 4 2 2 4 2 4 4 2 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1000] 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4
## [1037] 4 4 4 4 4 4 4 4 2 4 4 4 4 2 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4
## [1074] 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4
## [1111] 4 4 4 2 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 4 4 4 2 4 4 4 4 4 2 4 4 4 4
```

```
## [1148] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 2 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4
## [1185] 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1222] 4 4 4 4 4 4 4 2 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 4 4 4 4 4
## [1259] 4 4 2 4 2 4 4 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 4 4 4
## [1296] 4 4 2 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 2 4 4 4 4 4 4 4 4 4 4
## [1333] 4 4 4 4 2 4 2 4 4 4 4 4 2 4 4 4 4 4 2 2 2 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4
## [1370] 2 4 4 4 4 4 4 2 2 4 4 4 4 4 4 4 4 4 4 4 2 2 4 2 4 4 4 4 2 4 4 4 4 4 4 2 4 4
## [1407] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 4 4 4 2 4 4 4 4 2 4
## [1444] 4 4 4 4 4 4 4 2 4 4 4 4 2 4 4 4 2 4 4 2 4 4 4 4 4 2 2 4 4 4 2 4 4 4 4 4 4 4 4
## [1481] 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 4 2 4 4 4 4 4 4
## [1518] 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4
## [1555] 4 4 2 4 4 4 4 4 4 4 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 2 4 2 4 4 2 4 4 4
## [1592] 4 4 4 4 4 4 4 2 4 4 2 2 4 4 4 4 4 4 4 4 2 4 4 4 4 4 2 2 4 4 4 4 4 4 4 4 4 2 4 4
## [1629] 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1666] 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 4 2 4 4 4 4 4 4 4 4 2
## [1703] 4 4 2 2 4 4 4 4 2 4 4 2 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2
## [1740] 4 4 4 4 4 4 4 2 4 4 4 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 2 4 4 4 4 2 4 4 4 4 4 4 2
## [1777] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 2 2 2 4 2 2 4 4 2
## [1814] 4 4 4 4 1 4 4 2 4 4 4 2 4 4 2 4 4 2 2 2 4 2 2 2 2 4 2 4 4 4 2 4 2 4 4 2 2
## [1851] 2 2 4 2 2 2 2 4 2 4 4 2 2 4 2 4 2 2 4 4 4 4 2 4 4 2 2 2 2 4 4 4 4 2 2 2 4
## [1888] 4 4 2 2 4 4 4 4 4 2 2 4 4 4 2 4 4 4 4 4 4 2 4 4 2 2 2 4 4 4 4 4 4 2 2 4 4
## [1925] 4 2 2 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 2 2 4 4 4 4 2 2 4 2 2 2 2 2 2 4 4 2 2 2
## [1962] 2 4 2 4 4 4 2 4 2 4 4 4 2 2 4 4 2 2 4 2 4 4 4 4 2 2 4 4 2 4 2 2 4 2 4 2 4
## [1999] 4 4 4 2 2 4 2 4 4 4 4 4 2 4 4 2 4 4 4 4 4 4 4 2 2 2 4 2 2 4 2 4 4 2 4 2 2
## [2036] 2 4 4 4 4 4 2 2 2 4 4 4 4 2 4 2 2 4 4 4 4 2 4 2 2 4 4 4 4 4 2 4 2 2 4 4 2 2
## [2073] 2 2 2 4 4 2 2 4 2 4 4 2 2 4 4 4 2 2 4 2 4 4 4 4 2 2 4 4 4 2 2 2 2 2 2 2 2
## [2110] 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2147] 2 2 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2184] 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2
## [2221] 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 4 1 2 2 4 2 2 2 2 2 4 2 2 2 2 4 2 2 2
## [2258] 2 2 2 2 4 2 2 2 2 2 4 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 4 2 2 2 2
## [2295] 2 2 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2332] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 4 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2369] 4 2 2 2 2 2 2 2 4 2 4 4 2 2 2 2 2 2 1 2 4 4 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2
## [2406] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2443] 2 2 1 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2480] 4 4 2 2 4 2 2 2 2 4 2 2 2 2 2 2 2 4 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 2 4 2 2
## [2517] 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 4 2 2 2 2 2 2 2 2 2 2 2 2
## [2554] 2 2 2 2 2 2 2 2 4 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2591] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2628] 2 4 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2665] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 4 2 2 2 2 2 2 1 2 2
## [2702] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2
## [2739] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2
## [2776] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 4 2 2 2 4 2 2 2 2 2 2 2 2 2
## [2813] 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 4 2 2 2
## [2850] 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2
## [2887] 2 2 4 2 2 4 2 4 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2924] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2961] 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [2998] 2 2 2 2 2 1 2 1 4 2 2 2 2 1 2 2 2 4 2 2 1 1 2 2 2 2 2 2 2 2 2 4 1 2 2 2 2 2
```

```
## [3035] 2 1 2 2 2 2 2 1 4 2 2 2 2 2 4 2 2 2 2 4 2 2 2 2 2 4 2 2 4 1 2 1 2 2 2 4 1
## [3072] 2 2 2 1 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2 1 1 1 2 1 2 4 2 4 2 2 2 2 1 1 4 2 2
## [3109] 2 2 1 2 2 2 2 2 2 2 2 2 4 2 2 1 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2
## [3146] 2 2 2 4 4 4 2 2 1 2 2 2 4 2 2 1 2 1 4 2 1 4 2 2 1 2 2 2 2 1 4 2 2 2 2 1 2
## [3183] 4 4 2 2 2 2 1 1 2 1 4 2 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 1 4 2 1 2 2 2 2 2 2
## [3220] 4 2 2 2 2 2 2 2 1 4 2 2 2 1 4 1 2 1 2 2 2 1 2 2 4 1 2 1 1 2 1 2 2 2 2 4 2
## [3257] 2 2 2 4 2 2 2 4 2 1 1 2 2 2 1 4 2 2 2 1 2 4 1 2 2 1 1 4 2 1 2 2 2 2 2 2 2
## [3294] 4 2 2 4 2 4 1 2 1 2 2 2 1 2 4 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 4 1 2 2 4 4 1
## [3331] 2 2 4 2 2 1 2 2 4 2 1 2 1 2 2 2 2 2 2 2 1 2 4 1 2 2 2 2 4 2 2 2 2 1 1 2 4
## [3368] 2 2 2 2 2 4 1 2 2 2 2 2 2 1 2 4 2 2 2 2 2 2 4 1 4 2 2 4 2 4 2 4 2 1 4 2 1
## [3405] 4 2 4 2 2 2 2 2 4 2 2 2 2 2 1 2 2 2 4 2 2 2 1 2 2 2 2 2 2 1 2 2 4 2 1 2 1
## [3442] 2 4 2 4 4 2 4 1 2 2 2 1 4 1 4 2 2 4 2 2 2 2 2 4 2 1 2 2 2 4 2 2 2 1 2 2 2
## [3479] 2 2 1 2 2 2 4 4 2 2 2 1 2 1 2 1 2 2 2 1 2 2 2 4 2 2 2 2 2 2 2 1 2 2 1 2 2
## [3516] 2 4 2 2 2 2 2 4 1 2 2 2 2 2 2 2 2 4 2 1 2 2 4 2 2 1 2 2 2 4 1 2 2 4 2 2 2
## [3553] 2 2 2 4 2 1 2 2 1 4 2 2 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 4 2 2 2 2
## [3590] 2 2 4 2 2 2 1 2 2 1 2 1 2 2 2 2 1 4 1 1 1 2 1 2 2 2 1 2 4 2 2 2 2 1 1 2 2
## [3627] 2 2 4 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 1 1 1 2 2 1 1 1 2 4 2 1 2 2 1
## [3664] 2 4 2 2 2 1 2 2 4 2 2 2 2 1 2 2 2 4 2 2 2 4 1 2 1 4 2 2 2 2 2 4 2 1 2 2 4 2
## [3701] 4 2 1 2 4 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 1 4 1 2 2 2 2 2 2 2 1 2
## [3738] 2 2 4 2 2 1 4 2 2 2 1 2 2 2 4 4 2 4 4 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3775] 2 2 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 1 2 2 2 4 2 4 2 2 2 2 2 2 2 4 2 2 2 1 2
## [3812] 1 2 2 2 4 1 2 2 2 2 2 2 1 2 2 2 4 2 2 1 2 4 2 4 1 1 2 2 2 2 1 2 2 1 2 4 2
## [3849] 2 2 2 2 2 4 2 1 2 2 2 4 4 2 2 2 2 2 4 2 2 4 4 2 1 2 1 4 2 4 2 4 4 2 4 4 2
## [3886] 2 4 2 4 2 1 1 2 1 2 1 1 2 2 4 2 1 1 1 1 4 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 2
## [3923] 1 2 2 2 1 1 1 2 1 1 1 1 1 1 4 1 2 2 4 1 1 2 2 1 2 1 1 2 1 1 4 1 4 4 1 1 1
## [3960] 2 2 1 2 1 1 1 1 1 1 1 4 1 1 2 2 1 4 1 1 4 2 1 4 4 1 3 1 2 1 1 2 1 1 1 2 2
## [3997] 2 4 1 2 1 2 1 2 2 1 1 2 1 2 2 2 1 2 1 1 1 1 1 2 2 1 1 1 3 1 1 2 1 2 2 4 4
## [4034] 1 2 1 1 1 2 1 2 2 2 1 2 1 4 1 2 4 1 1 4 4 1 2 1 1 4 4 2 1 2 2 1 1 1 1 1 2
## [4071] 1 1 1 1 2 2 1 4 2 4 1 1 1 1 1 1 1 1 2 1 2 1 1 2 2 2 2 2 1 4 1 3 1 1 2 3 2
## [4108] 3 2 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1 2 1 1 1 4 2 1 1 1 2 1 1 2 2
## [4145] 4 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 2 1 1 1 2 1 2 1 1 2 1 2 1 1 1 2 1 1 2
## [4182] 1 1 1 2 1 1 2 2 1 1 1 1 1 2 2 2 3 2 4 4 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 4
## [4219] 1 1 2 1 1 1 1 2 1 1 1 1 1 1 4 4 1 1 1 1 1 1 1 1 1 1 2 1 4 1 2 1 1 1 1 1 2
## [4256] 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 4 1 1 4 3 1 1 1 1 1 1 1 1
## [4293] 3 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1
## [4330] 1 3 1 2 1 1 1 1 4 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 3 1 1 1 1 1 1 4
## [4367] 2 1 1 1 1 1 1 1 1 4 2 3 2 1 2 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1
## [4404] 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 3 1 1 1 1 2 1 1 1 1 3 1 1 1 1 1 1 1 2
## [4441] 1 4 1 1 1 2 1 1 1 1 1 1 2 1 1 1 4 1 1 1 3 1 3 1 1 1 1 1 1 1 4 1 1 1 1 1 1
## [4478] 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 4 1 2 3 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [4515] 1 1 1 1 1 1 1 2 1 1 3 1 1 2 1 1 1 1 1 1 1 4 3 1 1 1 2 1 1 1 1 1 2 1 1 1 2 4
## [4552] 1 1 1 1 1 1 1 2 4 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 4 1 1 2
## [4589] 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 3 2 1 1 1 1 1 1 1 1 1 1 1
## [4626] 1 1 1 1 2 2 2 1 2 1 2 1 1 1 1 1 1 4 2 1 1 2 1 4 1 1 1 1 2 1 1 1 1 1 1 1 1
## [4663] 4 2 4 1 2 4 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 4 1 2 1 1 2 1 4 1 1 2 1
## [4700] 1 4 1 1 4 1 2 1 1 1 2 2 1 1 1 1 1 1 1 4 1 2 1 1 3 1 1 2 1 1 1 1 1 1 2 1 2
## [4737] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [4774] 1 1 1 1 1 3 1 4 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 4 3 2 1 1 1 3 1 1
## [4811] 1 1 1 3 2 2 1 1 3 1 1 4 1 1 1 2 1 4 1 4 1 1 3 1 1 2 4 1 1 1 1 3 4 2 4 2 4
## [4848] 4 1 2 1 1 1 1 1 1 1 1 1 2 3 2 1 2 1 1 2 2 1 1 3 1 3 2 1 1 1 2 1 1 1 1 1 4
## [4885] 4 3 1 1 1 3 4 1 1 1 2 4 1 1 1 4 4 1 1 1 3 3 1 1 1 1 1 1 2 1 1 2 1 1 1 4 2
```

```
## [4922] 1 4 1 2 1 1 3 1 1 1 2 3 1 1 4 1 1 1 3 2 1 1 1 1 1 2 2 1 1 1 2 2 4 1 1 1 1
## [4959] 1 2 1 1 1 1 4 2 2 3 4 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 4 1 3 1 1 1 1 1 1 3
## [4996] 1 1 2 1 1 3 1 2 2 1 1 4 1 1 1 1 2 1 1 1 1 2 1 1 1 1 4 2 1 4 1 1 1 1 4 1 1
## [5033] 1 1 1 1 2 1 1 1 1 1 3 1 1 2 1 1 1 1 1 1 3 3 2 2 1 1 1 2 1 4 1 1 4 2 4 1 1
## [5070] 1 1 4 3 2 1 3 1 1 1 1 2 1 1 1 1 1 4 1 2 1 1 1 1 1 1 1 1 1 2 3 1 4 3 1 4 3
## [5107] 3 3 4 4 2 1 1 1 2 2 4 3 2 4 4 2 2 2 1 4 2 4 4 4 2 4 4 3 3 2 2 4 2 1 2 2 3
## [5144] 3 1 3 3 1 3 2 3 3 2 1 2 2 3 2 2 1 2 3 2 2 2 2 4 2 1 2 3 2 2 2 1 1 4 3 4 2
## [5181] 2 2 2 1 2 4 2 3 1 4 1 2 4 3 2 3 1 2 3 1 1 1 2 3 4 1 2 2 1 3 4 3 1 3 4 4 2
## [5218] 2 4 3 2 1 2 4 3 2 2 3 3 2 3 2 2 1 2 1 1 2 4 2 4 4 4 1 3 4 3 1 4 4 1 4 2 2
## [5255] 1 2 2 4 2 2 4 2 1 2 2 2 3 3 3 4 3 2 3 4 2 3 4 4 4 4 1 2 4 2 4 2 2 4 3 1 2
## [5292] 2 3 3 1 1 3 2 3 4 2 2 1 2 1 3 3 2 2 2 3 3 2 3 3 4 1 4 4 2 1 2 3 4 2 3 4 3
## [5329] 1 3 4 1 4 4 4 4 2 4 3 2 1 3 2 2 1 2 2 3 4 2 2 2 4 1 4 3 1 4 4 2 1 3 2 3 4
## [5366] 4 3 1 1 4 2 1 2 2 4 4 3 4 2 4 3 1 4 3 2 2 3 2 2 2 2 2 4 3 2 4 4 1 1 2 4 1
## [5403] 2 3 4 3 2 1 2 1 2 2 3 1 3 1 3 2 3 4 1 3 2 1 3 2 2 2 1 4 2 4 1 3 2 2 4 4 1
## [5440] 2 4 3 3 1 3 2 2 3 4 3 4 2 2 1 1 1 1 2 2 4 2 2 3 4 3 2 3 3 4 4 3 4 3 2 3 1
## [5477] 1 1 4 4 2 4 3 4 3 1 1 2 2 1 3 2 1 3 4 4 1 1 2 4 4 2 3 3 4 2 4 4 3 4 4 2 3
## [5514] 2 3 3 4 1 3 1 3 4 3 2 4 1 3 2 1 4 3 3 2 1 3 3 3 2 2 3 4 3 1 1 1 2 4 2 2 4
## [5551] 3 2 4 2 4 3 4 2 3 3 3 4 1 3 3 4 1 3 1 2 2 3 4 1 3 2 2 3 4 2 3 4 3 2 3 4 3
## [5588] 4 2 2 3 3 2 1 4 3 3 3 2 2 4 3 2 3 3 3 4 3 1 2 2 4 1 2 4 3 3 1 4 2 2 2 4 3
## [5625] 3 3 1 4 1 3 2 2 1 1 2 3 1 1 4 3 1 3 4 1 1 3 2 4 4 1 2 1 1 2 3 1 2 1 2 4 2
## [5662] 2 3 1 2 2 1 2 2 3 4 2 1 1 3 2 1 4 4 2 2 4 4 3 4 4 1 3 2 4 1 3 3 4 1 3 3 1
## [5699] 3 2 4 1 1 2 2 2 2 2 1 4 3 4 4 2 3 4 2 2 4 2 3 2 3 3 4 3 2 4 2 4 4 1 3 2 4
## [5736] 3 4 3 2 4 2 2 4 2 3 3 3 1 2 2 4 4 2 2 4 2 3 1 2 3 2 4 3 4 4 4 4 4 4 3 2 3
## [5773] 3 2 4 4 2 2 4 2 4 3 3 4 2 1 4 1 2 2 2 3 2 1 4 2 1 1 1 4 3 4 3 4 1 2 2 2 3
## [5810] 2 1 3 4 4 1 2 3 3 3 4 3 2 1 4 2 1 1 3 2 3 1 4 3 1 4 3 4 4 3 4 4 4 1 4 1 1
## [5847] 1 2 4 4 2 3 2 1 4 3 2 1 1 3 2 3 3 2 4 3 2 2 3 3 3 1 3 3 2 2 4 1 4 3 4 2 1
## [5884] 4 3 3 1 2 2 2 2 1 3 4 3 2 4 2 4 3 1 3 4 1 2 1 4 2 2 2 1 4 1 4 2 3 1 3 2 4
## [5921] 2 1 2 4 1 1 1 1 4 3 4 4 3 3 3 2 4 2 3 3 2 4 2 2 1 3 2 2 2 4 2 1 2 1 3 1 2
## [5958] 3 1 1 2 3 2 2 3 2 1 3 1 1 2 4 3 4 3 2 2 3 2 2 1 4 4 2 1 4 3 1 4 4 3 1 3 3
## [5995] 1 3 3 3 4 2 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6032] 3 3 3 3 3 3 3 3 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6069] 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6106] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6143] 3 4 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6180] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 2
## [6217] 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6254] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3
## [6291] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6328] 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6365] 3 3 3 4 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3
## [6402] 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 2 3 3
## [6439] 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6476] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3
## [6513] 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6550] 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6587] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6624] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6661] 3 2 3 3 2 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6698] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6735] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3
## [6772] 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

```
## [6809] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6846] 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [6883] 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 2 2 2 3 3 3 2 3 4 3 3 3 3 3
## [6920] 3 3 2 3 3 3 2 3 3 2 2 3 3 3 4 3 3 3 3 4 4 3 3 2 3 3 3 2 3 3 3 3 3 3 4 3 3
## [6957] 3 3 3 4 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 3 4 3 3 3 3 3 3 3 3 2 3 3 3 3 2 2 2
## [6994] 3 3 2 3 3 4 2 2 3 3 3 4 4 4 3 3 2 4 3 4 3 3 3 2 4 3 3 3 4 4 3 2 3 3 2 2 3 3
## [7031] 3 3 3 4 2 2 3 2 2 3 4 3 2 3 3 3 3 3 2 3 3 3 2 2 3 3 3 2 3 3 3 3 3 3 3 3 2 3 2
## [7068] 3 3 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 2 3 3 3 2 2 3 3 3 3 3 3 3 3 2 3 3
## [7105] 4 3 3 3 3 3 3 4 2 3 3 3 3 2 3 3 4 3 3 3 4 3 3 3 3 2 3 3 3 3 3 2 2 2 2 3 3 3
## [7142] 3 4 3 2 3 3 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2 3 3 2
## [7179] 3 3 3 3 2 2 4 3 2 3 3 2 3 3 3 3 3 3 2 3 3 3 3 2 3 4 3 2 2 3 3 3 3 3 4 3 3
## [7216] 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 4 2 3 3 2 2 2 2 3 3 2 3 3 3 3 2 2 2 3 3 3 3
## [7253] 2 3 2 3 4 3 3 3 3 2 3 3 4 3 2 2 3 3 3 3 4 3 3 3 3 4 3 4 3 4 2 3 3 4 2 3 3
## [7290] 3 3 2 3 2 2 3 4 3 3 3 2 2 2 3 3 3 4 2 3 3 4 3 4 3 3 2 4 3 3 3 4 3 3 3 2 3
## [7327] 2 3 2 4 2 3 3 3 3 4 2 2 3 3 3 3 3 2 3 4 3 2 3 3 2 2 3 2 2 4 3 3 3 2 4 2 2
## [7364] 3 3 3 3 2 4 2 3 2 3 3 2 3 2 2 2 3 3 3 3 3 3 3 4 2 3 2 3 4 2 3 3 3 3 3 3 2
## [7401] 3 4 4 3 3 3 2 3 2 3 3 3 2 2 2 3 2 2 3 2 3 3 3 2 3 3 2 3 3 3 3 3 3 2 3 2 3
## [7438] 3 2 3 2 2 3 3 2 3 4 2 3 3 2 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 3 2 3 2 3 3 3
## [7475] 3 3 3 3 3 3 3 2 2 2 3 4 3 2 2 3 3 3 2 3 3 3 3 3 3 3 4 3 4 3 3 3 3 3 3 3 3 3
## [7512] 2 3 3 3 4 3 2 3 3 3 3 3 2 2 4 3 2 3 3 2 3 2 4 2 3 4 2 2 3 3 4 4 2 2 3 3 3
## [7549] 2 2 3 4 3 4 3 3 2 3 3 2 4 2 2 3 3 4 3 2 3 3 3 2 3 3 3 3 3 2 3 3 3 3 4 3 3
## [7586] 4 3 2 3 3 3 4 3 3 4 3 3 3 2 3 3 4 3 2 3 2 3 3 3 3 2 4 3 2 3 3 3 3 2 3 3 3
## [7623] 3 3 4 3 3 4 2 3 3 2 4 4 3 3 2 2 3 3 3 4 2 3 2 4 3 3 3 3 4 4 4 2 3 3 3 3 3
## [7660] 2 4 3 3 2 3 2 2 3 3 3 3 3 2 3 2 2 3 3 3 3 3 4 3 4 3 3 4 3 3 2 2 3 3 2 3 3
## [7697] 3 3 3 4 2 3 2 2 3 2 2 4 3 3 2 2 3 4 3 2 3 2 2 3 3 2 2 3 2 4 4 3 2 3 2 3 3
## [7734] 3 3 3 3 2 2 3 4 4 2 3 2 4 2 4 2 4 4 3 3 2 4 3 3 3 3 3 2 3 2 2 4 3 2 3 3 3
## [7771] 4 3 3 2 3 3 3 4 4 3 2 3 3 2 2 3 3 3 3 2 3 4 2 3 3 3 4 3 4 3 3 3 2 3 2 3 2
## [7808] 3 2 3 3 4 2 3 2 4 3 3 2 3 3 2 3 3 4 3 2 3 3 2 2 2 2 3 2 2 3 2 3 3 3 2 3 2
## [7845] 3 2 2 3 2 3 3 3 3 3 3 4 3 3 3 2 2 4 3 3 3 3 3 2 3 3 3 4 2 4 3 3 3 3 3 2 3
## [7882] 3 3 4 3 3 3 2 2 3 3 4 3 3 2 2 3 3 3 2 3 4 3 3 3 2 2 3 4 3 2 2 2 2 2 3 2 2
## [7919] 3 4 2 3 2 2 3 3 3 3 2 3 4 3 2 3 2 3 3 3 2 4 2 2 2 3 3 2 2 2 3 3 3 3 3 2 2
## [7956] 4 3 2 2 4 3 4 3 3 3 2 2 4 3 3 3 2 2 4 3 4 4 4 2 3 2 3 2 3 3 2 2 2 3 2 3 3
## [7993] 2 3 3 2 3 2 3 3 2 2 2 4 2 3 3 3 3 3 2 2 2 3 2 2 4 3 4 4 4 3 3 3 3 3 2 2 3
## [8030] 3 3 4 3 3 3 2 4 2 3 2 3 2 3 2 2 2 3 3 3 3 3 3 4 2 3 2 2 2 2 3 3 3 3 3 4 3
## [8067] 2 3 2 2 2 3 2 3 2 2 2 2 2 3 3 3 3 3 3 2 4 2 3 2 3 3 3 3 3 3 2 4 3 3 4 2 3
## [8104] 2 2 2 2 2 2 3 4 2 4 2 3 2 2 3 3 3 2 2 2 3 2
##
## Within cluster sum of squares by cluster:
## [1]  44688.26  88745.46  44049.15 105532.66
##  (between_SS / total_SS =  47.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

The centroids for cluster are stored in KMean_clustering$centers and we display those

```
KMean_clustering$size
```

```
## [1] 1215 2574 1880 2456
```

```
KMean_clustering$centers
```

```
##         V2       V3       V4       V5       V6       V7       V8       V9
## 1 5.501235 3.498765 8.081481 2.000000 3.000000 2.000000 1.000000 1.000000
## 2 5.325175 3.654623 4.898601 2.806915 5.721834 1.918415 1.121212 1.113442
## 3 5.295745 4.461702 4.957447 2.055319 7.225532 2.000000 1.051064 2.838298
## 4 5.149837 3.685261 8.504479 2.486156 4.582248 2.000407 1.902687 1.519951
##        V10      V11      V12      V13      V14      V15      V16      V17
## 1 5.444444 1.000000 2.000000 2.000000 2.000000 4.183539 4.183539 1.000000
## 2 9.989510 2.452991 1.916472 2.876068 2.949106 6.917638 6.830614 1.000000
## 3 1.110638 2.838298 1.004255 2.527660 2.527660 7.927660 7.927660 1.000000
## 4 7.468241 1.822883 3.214577 2.590798 2.714577 8.983306 8.976792 1.000407
##       V18      V19      V20      V21      V22      V23
## 1 4.00000 2.000000 3.000000 2.000000 6.500412 3.041152
## 2 3.81352 2.184926 4.703963 4.226107 5.911033 1.914918
## 3 4.00000 2.161702 1.119149 8.987234 5.744681 3.844681
## 4 4.00285 2.156759 3.620928 3.921417 4.405130 2.996336
```

Now lets look at the correlation using table

```
KMean_corr_table <- table(mushrooms2$V1, KMean_clustering$cluster)
KMean_corr_table
```

```
##
##        1    2    3    4
##   1    0    0    0    1
##   2    0 2215  144 1849
##   3 1215  359 1736  606
```

We would use randIndex to calculate the clustering metrics. The value ranges from -1 (no agreement) to +1(perfect agreement)

```
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.2.3
```

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

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

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

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

```
randIndex(KMean_corr_table)
```

```
##       ARI
## 0.2717056
```

Here we can see that the value for randIndex is low. The dataset is not good for clustering.

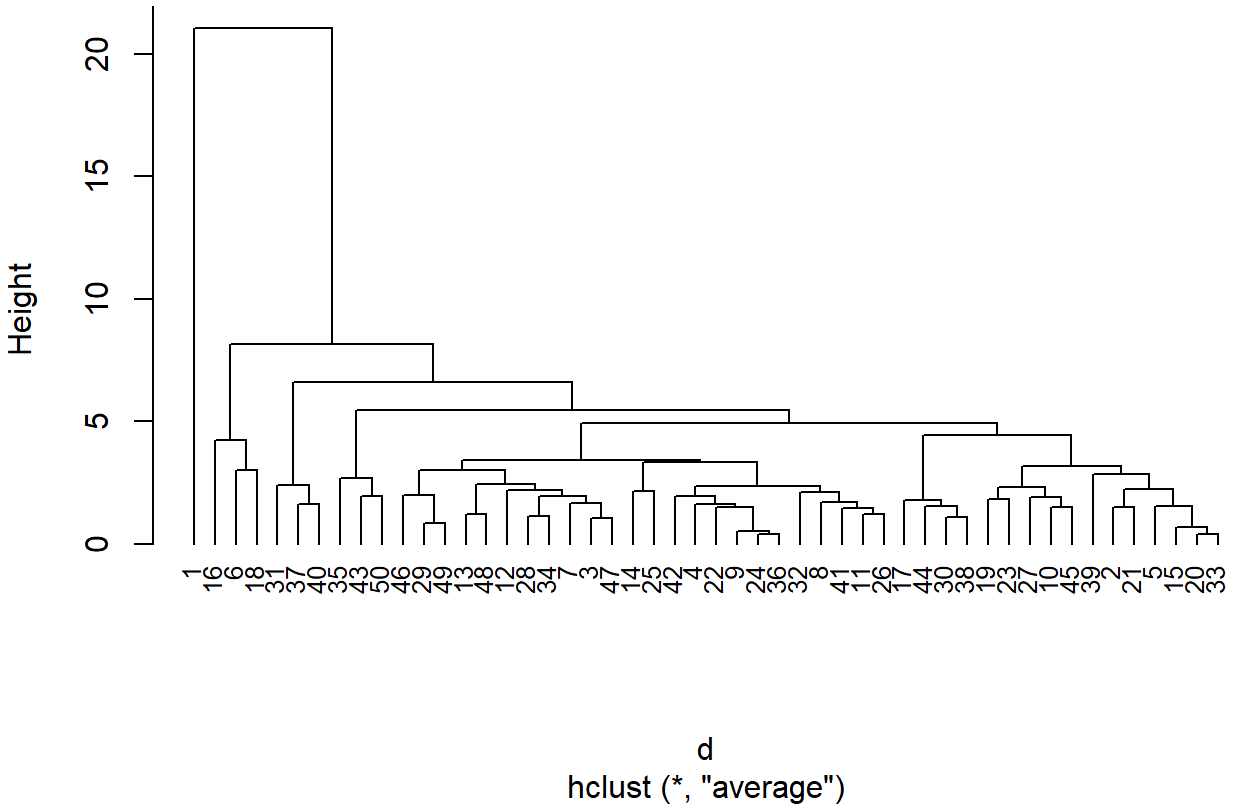
Hierarchical Clustering

First we would subset the mushroom dataset.Reduce the dataset to first 50 observations and then scale the data set to get optimum result.

```
mushroom_subset <- mushrooms3[1:50,]
mushroom_subset <- scale(mushroom_subset)
```

We Perform Hierarchical clustering by first finding the Euclidean distances between the observations by using average linkage using dist() and then we perform the Hierarchical clustering using hclust. We can then plot the cluster. The resultant plot is a dendogram for first 50 observations in the mushroom dataset

```
d <- dist(mushroom_subset)
fit.average <- hclust(d, method="average")
plot(fit.average, hang=-1, cex=.8,
     main="Hierarchical Clustering")
```



Now we will cut the dendogram and trying different cuts from 3 to 11. The first column in the mushrooms2 contains the classes of mushroom. We create a table for the first 50 values based on classes from mushroom2 and cluster_cut

```
for (c in 3:11){
  cluster_cut <- cutree(fit.average, c)
  table_cut <- table(cluster_cut, mushrooms2[1:50,1:1])
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand index = ", ri))
}
```

```
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0 34 12
##           3  0  3  0
## [1] "cut= 3 Rand index =  0.0367982387167792"
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0 31 12
##           3  0  3  0
##           4  0  3  0
## [1] "cut= 4 Rand index =  -0.0222621565415505"
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0 28 12
##           3  0  3  0
##           4  0  3  0
##           5  0  3  0
## [1] "cut= 5 Rand index =  -0.0638951377503252"
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0  4 12
##           3  0 24  0
##           4  0  3  0
##           5  0  3  0
##           6  0  3  0
## [1] "cut= 6 Rand index =  0.352204324263506"
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0  0 12
##           3  0 24  0
##           4  0  3  0
##           5  0  4  0
##           6  0  3  0
##           7  0  3  0
## [1] "cut= 7 Rand index =  0.43383285451769"
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0  0 12
##           3  0 24  0
##           4  0  2  0
##           5  0  1  0
##           6  0  4  0
##           7  0  3  0
##           8  0  3  0
## [1] "cut= 8 Rand index =  0.4311484193606"
```

```
##
## cluster_cut  1  2  3
##           1  1  0  0
##           2  0  0 12
##           3  0 11  0
##           4  0 13  0
##           5  0  2  0
##           6  0  1  0
##           7  0  4  0
##           8  0  3  0
##           9  0  3  0
## [1] "cut= 9 Rand index =  0.24707339675095"
##
## cluster_cut  1  2  3
##          1   1  0  0
##          2   0  0 12
##          3   0 11  0
##          4   0 11  0
##          5   0  2  0
##          6   0  2  0
##          7   0  1  0
##          8   0  4  0
##          9   0  3  0
##          10  0  3  0
## [1] "cut= 10 Rand index =  0.220066017455861"
##
## cluster_cut  1  2  3
##          1   1  0  0
##          2   0  0  7
##          3   0 11  0
##          4   0 11  0
##          5   0  2  0
##          6   0  0  5
##          7   0  2  0
##          8   0  1  0
##          9   0  4  0
##          10  0  3  0
##          11  0  3  0
## [1] "cut= 11 Rand index =  0.177780621295752"
```

Cut 7, 8 and 6 gives us the best correspondence with the classes of mushroom present in the dataset.

Model Based Clustering

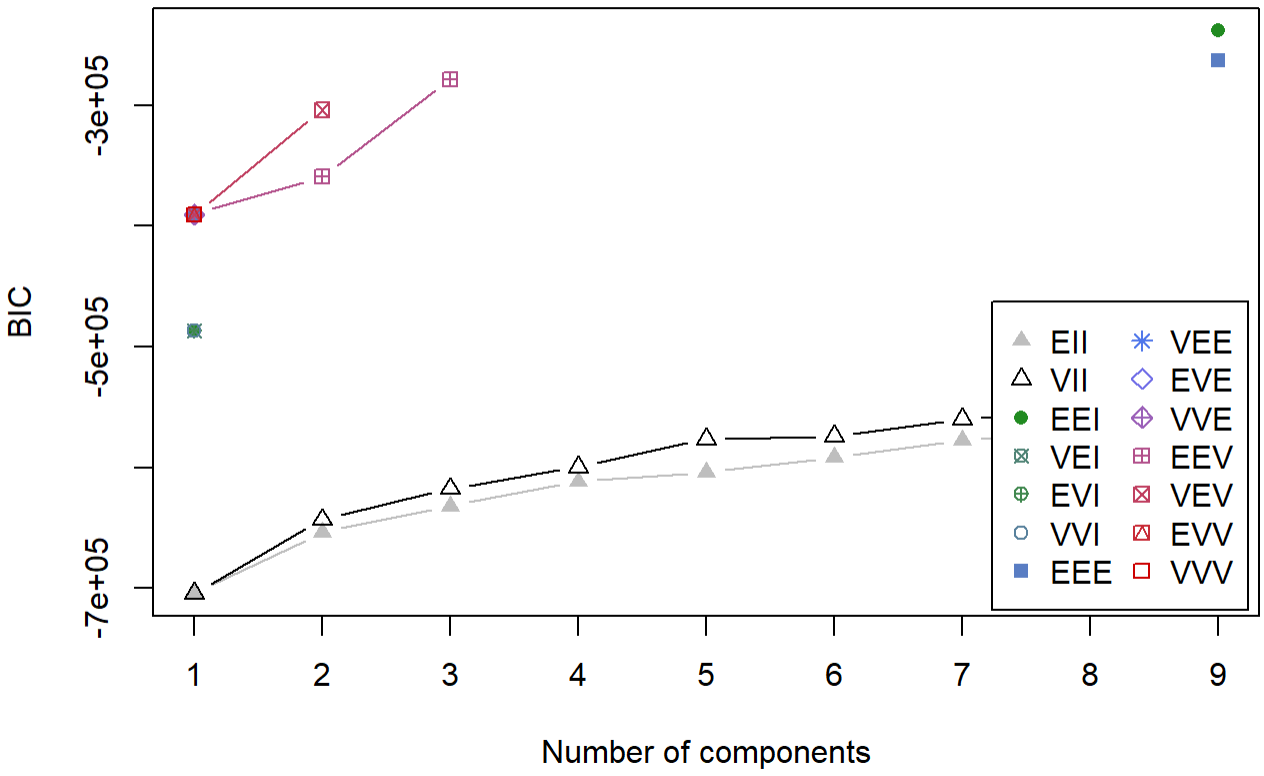
Model Based clustering applies the maximum likelihood estimation and Bayes criteria to find the number of cluster and most likely method. This is done by assuming variety of data models. We can use the Mclust() to perform Model Based Clustering

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.2.3
```

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

```
model_based_clustering <- Mclust(mushrooms3)
plot(model_based_clustering, what="BIC") # plot results
```




```
MBC_corr_table <- table(mushrooms2$V1, model_based_clustering$classification)
MBC_corr_table
```

```
##
##        1    2    3    4    5    6    7    8    9
##   1    0    0    1    0    0    0    0    0    0
##   2  912    0  768  528  800   96  912  192    0
##   3  288  576    0   72  272 1556    0    0 1152
```

```
randIndex(MBC_corr_table)
```

```
##       ARI
## 0.1849217
```

Comparing the results of Different clustering algorithm

randIndex was used to calculate the metrics for the clustering algorithms. The ARI results we got from KMeans clustering was 0.2717056. The ARI result from the Hierarchical Clustering is based on different cuts. Specifically the cut at 7 yielded the result of 0.4338. The ARI result from Model Base Clustering was 0.12. The problem with Hierarchical Clustering is that it places each observation into its own cluster, calculates the distance between the clusters and combine the two closest clusters. This is repeated until convergence. Since we had 8000 observations the hierarchical clustering was not giving any valuable information. So we performed the hierarchical clustering on the first 50 observations, so we can't consider the results from hierarchical clustering. We can clearly see that hierarchical clustering falls short on large sets of data. It would be appropriate to use KMeans clustering. Even for KMeans the ARI was low. We can conclude that this dataset is not suitable for clustering