Assignment 5

Sungjoo Cho and Kate Lamoreaux

2023-12-05

GitHub link:

https://github.com/katelmrx/cho-lamoreaux-a1.git

Packages

```
library(censusapi)
library(tidyverse)
library(magrittr)
library(factoextra)
library(base)
library(RSocrata)
library(ggmap)
library(lubridate)
library(corrplot)
library(dplyr)
library(knitr)
```

Exploring ACS Data

```
cs_key <- readLines("cs_key.TeX")
```

```
# Data from the American Community Survey (ACS)
acs_il_c <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME", "B01003_001E", "B19013_001E", "B19301_001E"),
                      region = "county:*",
                      regionin = "state:17",
                      key = cs_key) %>%
  rename(pop = B01003_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
head(acs_il_c)
```

```
##   state county                      NAME    pop hh_income income
## 1    17    067   Hancock County, Illinois  18633     50077  25647
```

```
## 2    17    063    Grundy County, Illinois  50338     67162  30232
## 3    17    091  Kankakee County, Illinois 111493     54697  25111
## 4    17    043    DuPage County, Illinois 930514     81521  40547
## 5    17    003 Alexander County, Illinois   7051     29071  16067
## 6    17    129    Menard County, Illinois  12576     60420  31323
```

```
# Pull map data for Illinois into a data frame.
il_map <- map_data("county", region = "illinois")
head(il_map)
```

```
##        long      lat group order   region subregion
## 1 -91.49563 40.21018     1     1 illinois     adams
## 2 -90.91121 40.19299     1     2 illinois     adams
## 3 -90.91121 40.19299     1     3 illinois     adams
## 4 -90.91121 40.10704     1     4 illinois     adams
## 5 -90.91121 39.83775     1     5 illinois     adams
## 6 -90.91694 39.75754     1     6 illinois     adams
```

Join the ACS data with the map data. Note that il_map has a column subregion which includes county names. We need a corresponding variable in the ACS data to join both data sets. This needs some transformations, among which the function tolower() might be useful. Call the joined data acs_map.

```
# creating county_name variable in acs_il
acs_il_c$subregion <- gsub(", .*", "", acs_il_c$NAME)
acs_il_c$subregion <- gsub("County", "", acs_il_c$subregion)
acs_il_c$subregion <- trimws(acs_il_c$subregion, "right")
acs_il_c$subregion <- tolower(acs_il_c$subregion)

# join
acs_map <- inner_join(acs_il_c, il_map, by=join_by("subregion" == "subregion"))
head(acs_map)
```

```
##   state county                     NAME   pop hh_income income subregion
## 1    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 2    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 3    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 4    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 5    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 6    17    067 Hancock County, Illinois 18633     50077  25647   hancock
##        long      lat group order   region
## 1 -91.18623 40.63417    34   573 illinois
## 2 -90.89976 40.63417    34   574 illinois
## 3 -90.91121 40.27893    34   575 illinois
## 4 -90.91121 40.19299    34   576 illinois
## 5 -91.49563 40.21018    34   577 illinois
## 6 -91.48990 40.25029    34   578 illinois
```

```
# number of rows
dim(acs_il_c)
```

```
## [1] 102   7
```

```
dim(il_map)
```

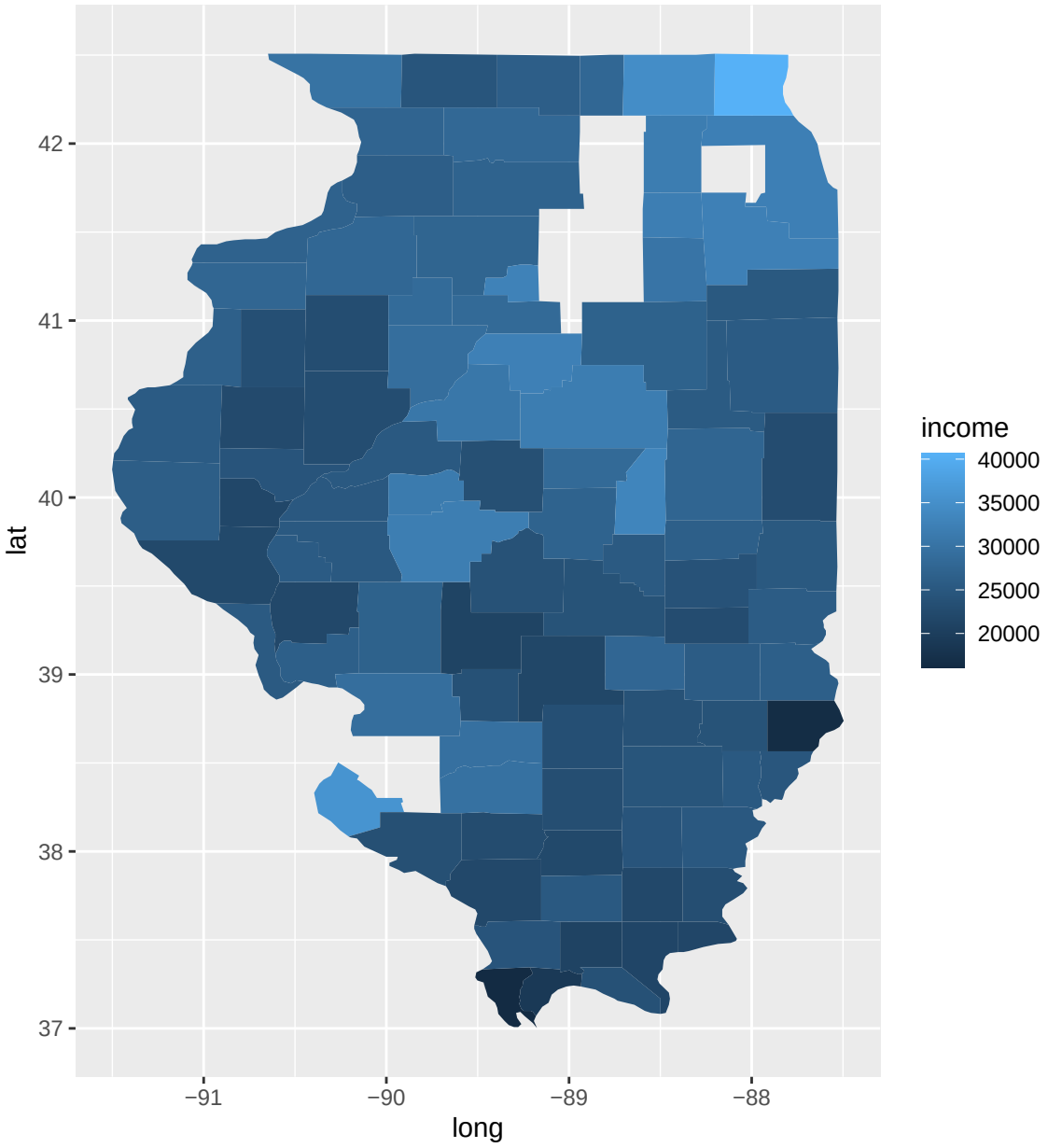
```
## [1] 1697    6
```

```
dim(acs_map)
```

```
## [1] 1642   12
```

After you do this, plot a map of Illinois with Counties colored by per capita income.

```
ggplot(acs_map) +
geom_polygon(aes(x = long, y = lat, group = group, fill = income))
```



Hierarchical Clustering

We want to find clusters of counties that are similar in their population, average household income and per capita income. First, clean the data so that you have the appropriate variables to use for clustering.

```
head(acs_map)
```

```
##   state county                     NAME   pop hh_income income subregion
## 1    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 2    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 3    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 4    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 5    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 6    17    067 Hancock County, Illinois 18633     50077  25647   hancock
##        long      lat group order   region
## 1 -91.18623 40.63417    34   573 illinois
## 2 -90.89976 40.63417    34   574 illinois
## 3 -90.91121 40.27893    34   575 illinois
## 4 -90.91121 40.19299    34   576 illinois
## 5 -91.49563 40.21018    34   577 illinois
## 6 -91.48990 40.25029    34   578 illinois
```

```
# select pop, hh_income, and income variables
hclust_data <- acs_map %>%
  select(pop, hh_income, income)

# cleaned data
head(hclust_data)
```

```
##     pop hh_income income
## 1 18633     50077  25647
## 2 18633     50077  25647
## 3 18633     50077  25647
## 4 18633     50077  25647
## 5 18633     50077  25647
## 6 18633     50077  25647
```

Next, create the distance matrix of the cleaned data. This distance matrix can be used to cluster counties, e.g. using the ward method.

```
# distance matrix of the cleaned data
hclust_d <- dist(hclust_data)
as.matrix(hclust_data)[1:10,]
```

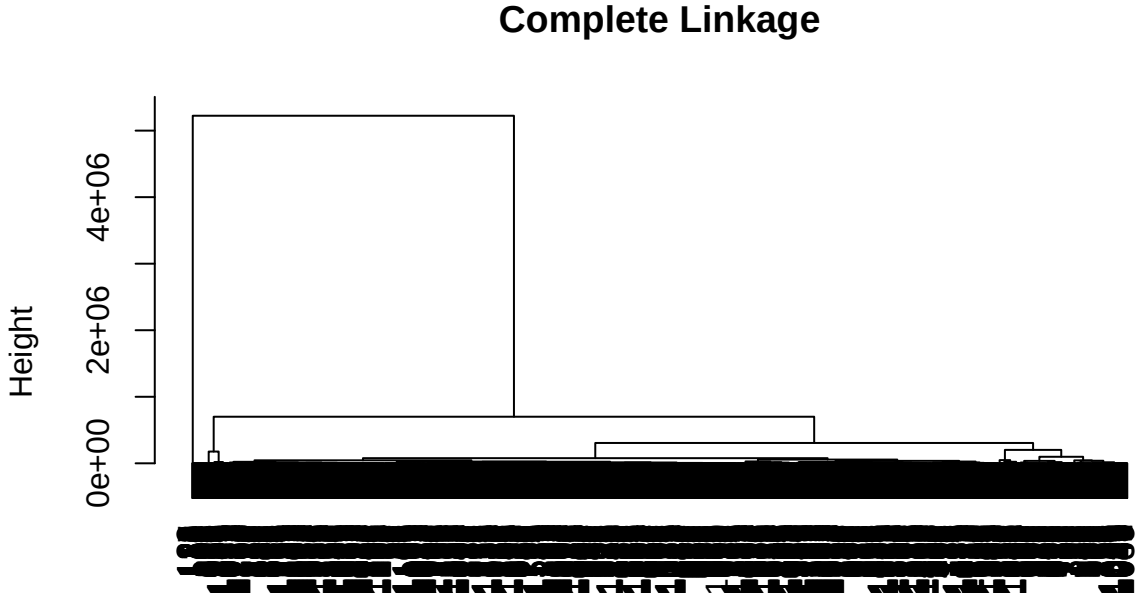
```
##         pop hh_income income
##  [1,] 18633     50077  25647
##  [2,] 18633     50077  25647
##  [3,] 18633     50077  25647
##  [4,] 18633     50077  25647
##  [5,] 18633     50077  25647
##  [6,] 18633     50077  25647
##  [7,] 18633     50077  25647
```

```
##  [8,] 18633     50077  25647
##  [9,] 18633     50077  25647
## [10,] 18633     50077  25647
```

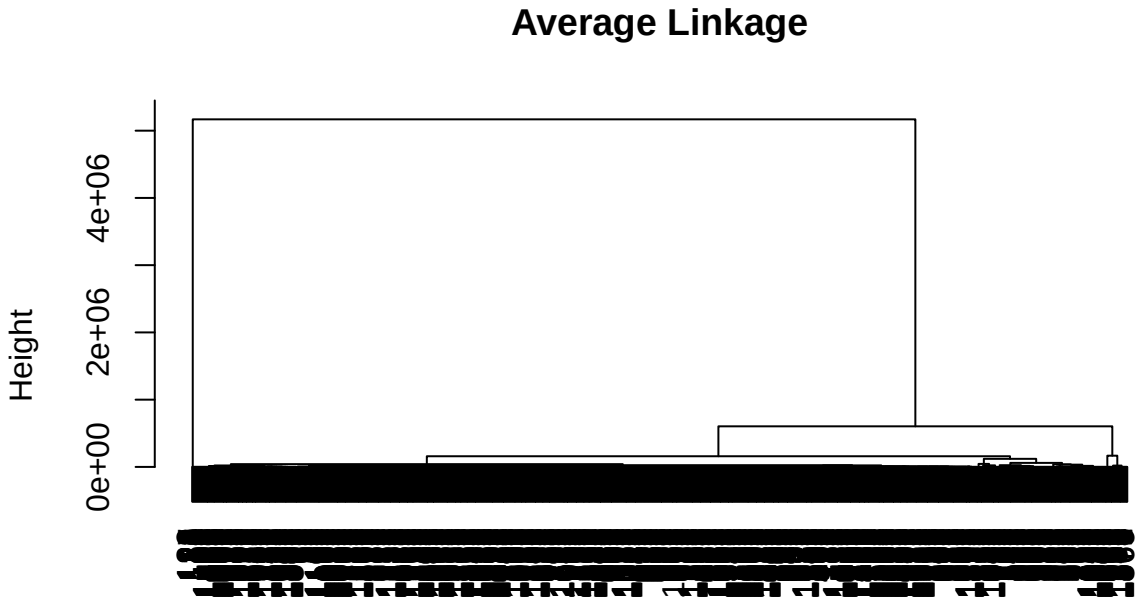
```r
# clustering counties using the ward method
hc_ward <- hclust(hclust_d, method = "ward.D2")

# clustering counties using the complete method
hc_complete <- hclust(hclust_d, method = "complete")

# clustering counties using the average method
hc_average <- hclust(hclust_d, method = "average")
```

Plot the dendrogram to find a reasonable number of clusters. Draw boxes around the clusters of your cluster solution.

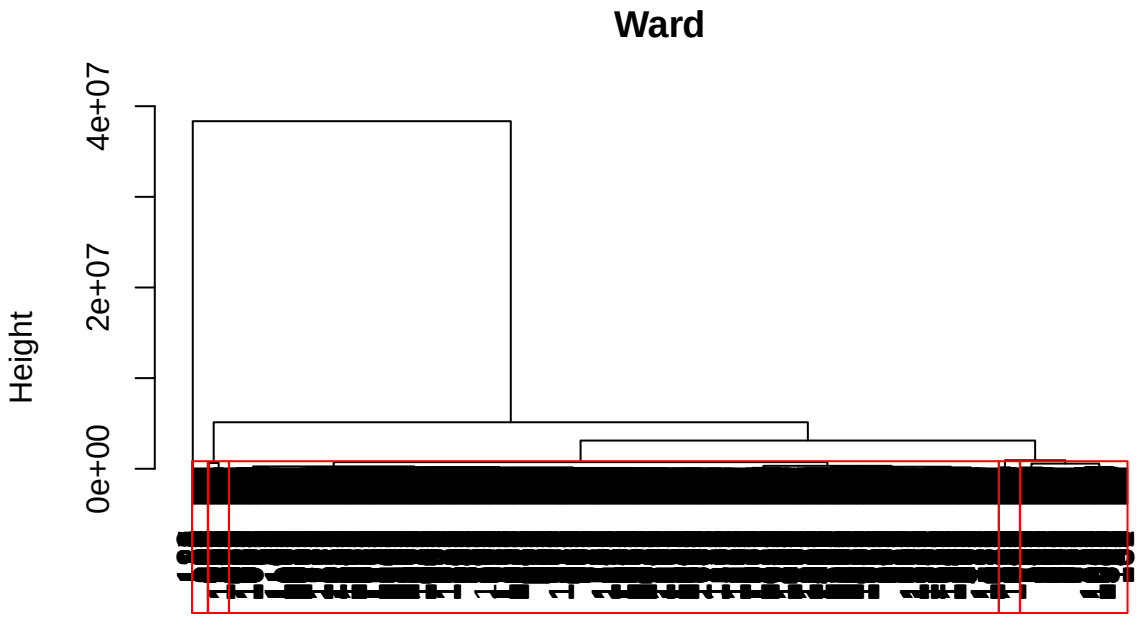
```r
# complete method
plot(hc_complete, main = "Complete Linkage", xlab = "", sub = "")
```




```r
# average method
plot(hc_average, main = "Average Linkage", xlab = "", sub = "")
```




```
# ward method
plot(hc_ward, main = "Ward", xlab = "", sub = "")

rect.hclust(hc_ward,
            k = 5,
            border = "red")
```



We want to create five clusters based on the `hc_ward` object.

```
cutree(hc_ward, 5)
```

```
##   [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 2 2 2 2 2 2 2 2
##  [38] 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 1 1
##  [75] 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 3 3 3 3 3 3 3 3 3
```

```
##  [112] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1
##  [149] 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 1 1
##  [186] 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 4 4 4 4 4
##  [223] 4 4 4 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
##  [260] 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 1 1
##  [297] 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1
##  [371] 1 1 1 1 1 1 2 2 2 2 2 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 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
##  [445] 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 1 1
##  [482] 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 1 1
##  [519] 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 4 4 4 4 4 4 4 4 4 4
##  [556] 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
##  [593] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [630] 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 1 1 1
##  [667] 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 2 2
##  [704] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 1 1 1 1 1 1 1 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
##  [778] 2 2 2 2 2 2 2 2 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
##  [815] 1 1 1 1 5 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [852] 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 1 1 1
##  [889] 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 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 5
##  [963] 5 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 1 1
## [1000] 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 1 1 1
## [1037] 1 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 4 4 4 4 4 4 1 1 1 1
## [1074] 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 1 1 1
## [1111] 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 1 1 1
## [1148] 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 1 1 1
## [1185] 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 1 1 1
## [1222] 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 1 1 1
## [1259] 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 1 1 1
## [1296] 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 1 1 1
## [1333] 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 1 1 1
## [1370] 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 1 1 1
## [1407] 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 1 1 1
## [1444] 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 1 1 1
## [1481] 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 1 1 1
## [1518] 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 1 1 1
## [1555] 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 1 1 1 1 1
## [1592] 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 1 1 1
## [1629] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

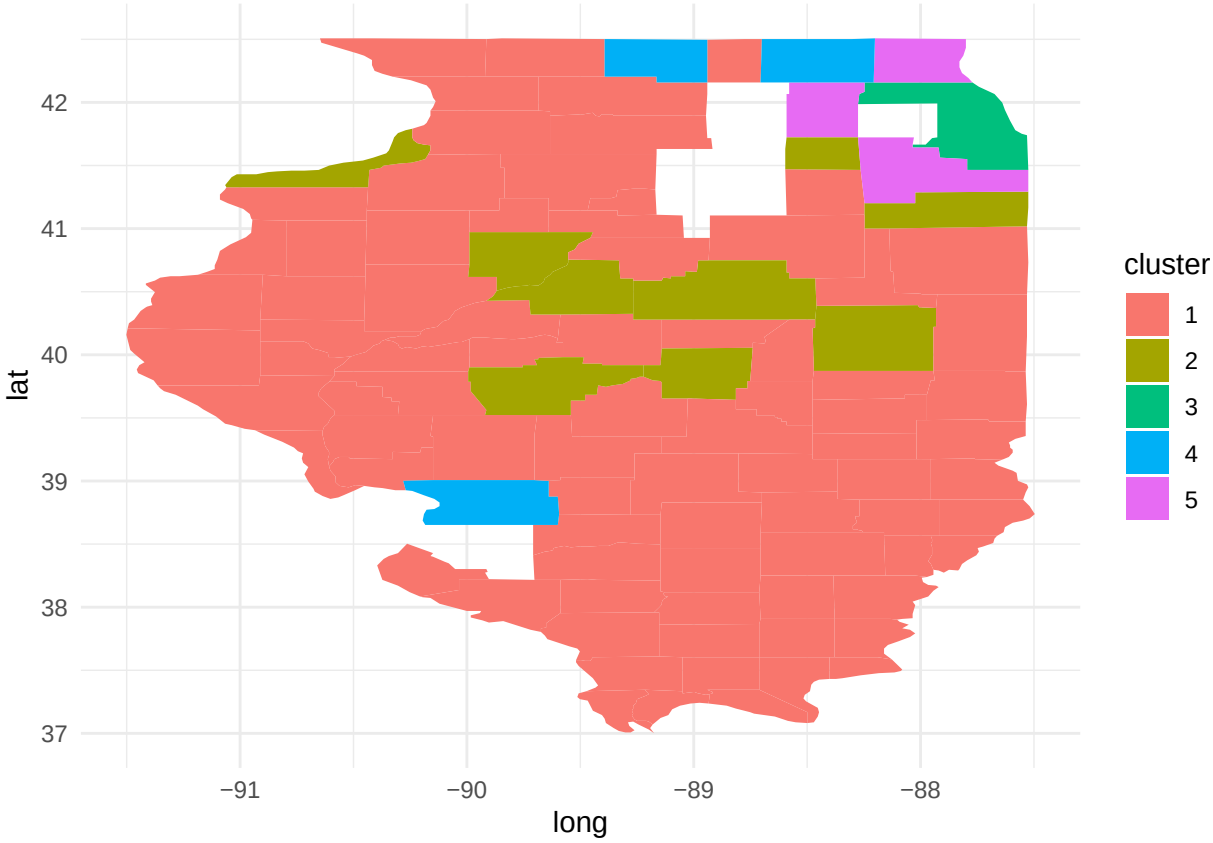
Visualize the county clusters on a map. For this task, create a new `acs_map` object that now also includes cluster membership as a new column. This column should be called `cluster`.

```
# adding cluster membership variable
acs_map <- acs_map %>%
  mutate(cluster = cutree(hc_ward, 5))

# new acs_map
head(acs_map)
```

```
##   state county                    NAME   pop hh_income income subregion
```

```
## 1    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 2    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 3    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 4    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 5    17    067 Hancock County, Illinois 18633     50077  25647   hancock
## 6    17    067 Hancock County, Illinois 18633     50077  25647   hancock
##        long      lat group order   region cluster
## 1 -91.18623 40.63417    34   573 illinois       1
## 2 -90.89976 40.63417    34   574 illinois       1
## 3 -90.91121 40.27893    34   575 illinois       1
## 4 -90.91121 40.19299    34   576 illinois       1
## 5 -91.49563 40.21018    34   577 illinois       1
## 6 -91.48990 40.25029    34   578 illinois       1
```

```
# visualize the county clusters on map
ggplot(acs_map) +geom_polygon(aes(x=long, y=lat, group=group, fill=factor(cluster))) +
  theme_minimal( ) + labs(fill="cluster")
```



Census Tracts

For the next section we need ACS data on a census tract level. We use the same variables as before.

```
acs_il_t <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME", "B01003_001E", "B19013_001E", "B19301_001E"),
```

```
                      region = "tract:*",
                      regionin = "state:17",
                      key = cs_key) %>%
  mutate_all(list(~ ifelse(.==-666666666, NA, .))) %>%
  rename(pop = B01003_001E,
         hh_income = B19013_001E,
         income = B19301_001E)

acs_il_t <- na.omit(acs_il_t)
head(acs_il_t)
```

```
##   state county  tract                                          NAME  pop
## 1    17    031 806002 Census Tract 8060.02, Cook County, Illinois 7304
## 2    17    031 806003 Census Tract 8060.03, Cook County, Illinois 7577
## 3    17    031 806400    Census Tract 8064, Cook County, Illinois 2684
## 4    17    031 806501 Census Tract 8065.01, Cook County, Illinois 2590
## 5    17    031 750600    Census Tract 7506, Cook County, Illinois 3594
## 6    17    031 310200    Census Tract 3102, Cook County, Illinois 1521
##   hh_income income
## 1     56975  23750
## 2     53769  25016
## 3     62750  30154
## 4     53583  20282
## 5     40125  18347
## 6     63250  31403
```

k-Means

As before, clean our data for clustering census tracts based on population, average household income and per capita income.

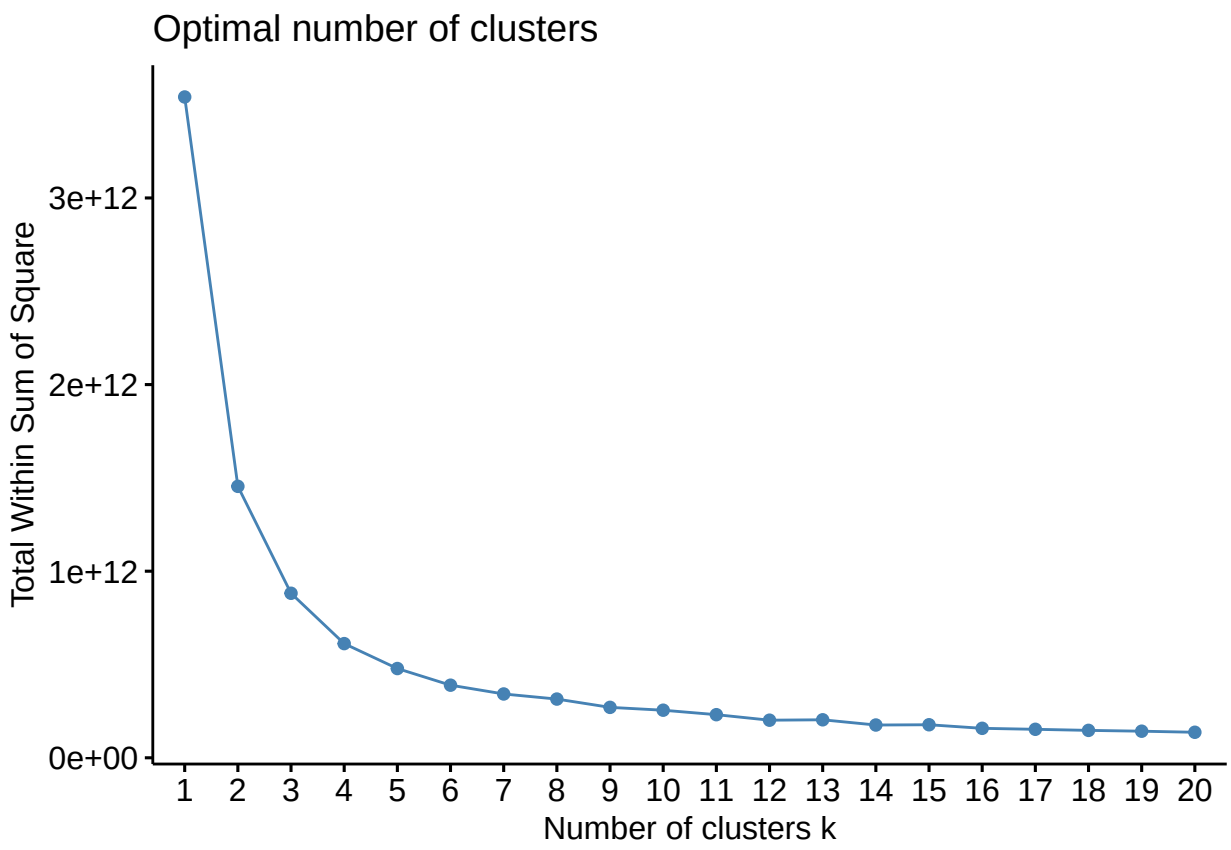
```
# select pop, hh_income, income
acs_il_t_cleaned <- acs_il_t %>%
  select(pop, hh_income, income) %>%
  na.omit()

# new dataset
head(acs_il_t_cleaned)
```

```
##    pop hh_income income
## 1 7304     56975  23750
## 2 7577     53769  25016
## 3 2684     62750  30154
## 4 2590     53583  20282
## 5 3594     40125  18347
## 6 1521     63250  31403
```

Since we want to use K Means in this section, we start by determining the optimal number of K that results in Clusters with low within but high between variation. Plot within cluster sums of squares for a range of K (e.g. up to 20).

- We choose k = 6 as the optimal number of clusters.

```r
# finding the optimal number of K
fviz_nbclust(acs_il_t_cleaned, kmeans, method = "wss", k.max = 20)
```



Run `kmeans()` for the optimal number of clusters based on the plot above.

```r
# kmeans
set.seed(24601)
(km <- kmeans(acs_il_t_cleaned, 6, nstart = 20))
```

```
## K-means clustering with 6 clusters of sizes 568, 828, 454, 228, 73, 958
##
## Cluster means:
##        pop hh_income   income
## 1 3023.996  27265.57 15527.79
## 2 4357.746  64282.81 31434.40
## 3 5107.751  87549.29 41192.31
## 4 4535.430 112380.92 59990.76
## 5 4126.795 163369.92 78493.00
## 6 4029.991  46117.11 23254.56
##
## Clustering vector:
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
##    2    6    2    6    6    2    1    2    1    1    6    6    5    5    5    5
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
##    4    6    4    3    2    3    2    1    2    2    2    2    1    2    1    2
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
```

```
##   6   1   6   3   2   6   6   2   3   2   6   3   2   3   2   2
##  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
##   3   3   6   6   3   2   2   2   2   1   6   1   1   1   1   1
##  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   3   3   1   1   1   6   6   2   2   1   6   6   6   1   1   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
##   1   1   3   3   6   3   3   2   2   3   4   4   4   3   4   4
##  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112
##   2   6   6   2   6   4   5   5   4   3   4   5   4   5   4   5
## 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
##   2   3   2   3   2   3   3   6   3   4   5   4   3   3   6   2
## 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   3   1   1   1   6   1   6   1   6   3   6   1   1   6   6
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   4   1   6   6   2   6   2   6   6   6   6   6   1   6   2   1
## 161 162 163 164 165 166 167 168 169 170 171 172 173 175 176 177
##   1   6   1   2   1   2   6   1   1   4   2   2   1   1   1   1
## 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193
##   6   6   6   6   2   2   6   1   6   6   6   6   6   6   6   2
## 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
##   6   6   6   6   2   1   1   2   2   2   2   6   6   2   6   2
## 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
##   3   1   2   2   2   2   6   2   6   6   1   1   1   1   4   6
## 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241
##   3   2   1   2   2   6   6   6   1   1   6   6   1   6   1   2
## 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257
##   2   2   6   2   6   2   6   2   2   2   6   2   6   6   6   2
## 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273
##   6   6   1   1   6   2   2   2   3   3   6   4   2   2   4   2
## 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289
##   4   2   2   4   3   3   2   3   4   5   2   2   2   2   3   3
## 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
##   3   3   4   5   3   4   2   2   6   6   6   2   6   6   6   6
## 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321
##   6   6   6   6   6   2   2   2   2   6   6   6   2   6   6   3
## 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337
##   3   2   2   3   4   2   4   3   2   3   2   2   2   3   6   6
## 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353
##   2   2   3   3   3   5   5   5   4   1   1   2   3   3   2   2
## 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369
##   3   3   6   1   1   3   2   4   2   6   6   2   2   6   2   6
## 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385
##   2   6   6   6   6   6   6   1   6   2   2   6   6   1   2   1
## 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401
##   6   6   2   2   6   2   6   2   1   2   6   1   1   4   6   3
## 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
##   4   2   1   6   2   1   1   2   2   3   3   3   3   1   1   1
## 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433
##   4   1   3   4   4   4   3   3   6   2   4   2   1   2   1   3
## 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449
##   4   6   6   6   1   6   6   6   2   6   1   6   1   2   6   6
## 450 451 452 453 454 455 456 457 458 459 460 461 462 464 465 466
##   6   6   2   1   2   2   6   5   3   5   5   5   4   4   2   1
## 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482
```

```
##   6   6   1   2   4   6   2   6   2   2   2   6   6   2   6   6
##  483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498
##   6   6   5   2   6   2   2   2   2   3   1   6   6   6   6   3
##  499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514
##   2   3   3   3   4   4   4   3   3   3   3   6   3   2   4   2
##  515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530
##   3   4   3   2   6   2   2   3   3   2   1   1   3   3   4   4
##  531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546
##   6   4   3   4   6   1   1   6   6   6   1   1   1   1   1   4
##  547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562
##   4   6   5   4   4   4   3   4   4   1   2   6   6   6   6   6
##  563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578
##   2   2   6   2   3   6   3   3   4   3   3   3   6   6   2   4
##  579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594
##   4   4   5   4   4   5   3   4   4   3   5   4   2   2   2   6
##  595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610
##   1   6   1   1   4   3   6   4   3   6   3   2   6   6   2   2
##  611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626
##   1   1   2   2   3   6   2   2   1   1   6   6   6   2   1   6
##  627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642
##   6   1   1   1   2   6   6   3   1   1   1   1   6   1   6   6
##  643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658
##   6   1   6   2   1   1   1   1   2   4   6   1   1   6   2   6
##  659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674
##   4   6   4   1   2   6   6   2   1   6   6   6   2   6   6   2
##  675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
##   6   1   1   6   3   1   2   2   1   6   6   6   4   3   3   1
##  691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706
##   4   4   3   2   3   5   3   4   3   3   2   2   2   2   3   6
##  707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722
##   3   3   3   6   4   3   4   3   4   4   2   6   6   2   2   2
##  723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738
##   4   6   4   1   2   2   6   2   2   2   2   2   2   2   2   6
##  739 741 742 743 744 746 747 748 749 750 751 752 753 754 755 756
##   2   2   6   2   2   2   6   2   6   1   6   3   2   1   6   1
##  757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772
##   1   2   1   6   2   2   6   4   2   4   1   6   2   6   2   5
##  773 774 775 776 777 778 779 780 782 783 784 785 786 787 788 789
##   6   6   6   6   6   1   6   6   2   1   6   6   1   1   6   1
##  790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805
##   6   3   6   1   6   6   1   3   3   4   4   4   3   4   4   4
##  806 807 808 809 810 811 812 813 815 816 817 818 819 820 821 822
##   3   3   2   4   1   4   3   6   6   2   2   1   6   6   2   6
##  823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838
##   1   3   5   1   6   6   1   1   6   2   1   2   2   6   2   6
##  839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854
##   2   2   6   6   2   2   2   2   6   6   6   2   6   6   6   6
##  855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870
##   2   6   6   6   1   1   1   2   2   6   2   2   3   3   3   3
##  871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886
##   6   2   4   6   6   2   4   4   4   3   2   2   2   4   2   6
##  887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902
##   6   2   4   4   4   2   3   2   2   3   3   4   3   4   2   2
##  903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918
```

```
##    6    3    3    2    2    4    3    6    6    1    6    2    3    2    2    6
##  919  920  921  922  923  924  925  926  927  928  929  930  931  932  933  934
##    6    2    2    6    6    1    6    6    6    6    1    6    1    2    6    3
##  935  936  937  938  939  940  941  942  943  944  945  946  947  948  949  950
##    6    3    6    6    1    1    6    1    6    1    6    6    6    6    6    2
##  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966
##    2    2    2    6    6    3    1    6    6    6    6    2    2    3    3    3
##  967  968  969  970  971  972  973  974  975  976  977  978  979  980  981  982
##    3    3    3    3    4    1    1    6    2    2    6    6    2    6    2    6
##  983  984  985  986  987  988  989  990  991  992  993  994  995  996  997  998
##    6    6    6    2    2    2    3    3    2    3    3    3    2    3    3    1
##  999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014
##    6    1    1    1    1    1    3    2    1    4    4    6    1    2    1    1
## 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030
##    1    1    6    1    1    1    1    4    2    2    6    5    4    1    6    6
## 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046
##    5    5    4    4    2    3    6    2    2    6    4    5    2    3    2    2
## 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062
##    6    6    6    6    6    6    6    6    1    6    6    6    2    6    6    6
## 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078
##    6    6    1    6    1    1    1    1    2    3    2    3    3    3    6    6
## 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094
##    6    6    6    2    6    2    2    1    4    3    3    3    2    3    2    2
## 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110
##    3    6    2    6    6    6    2    3    5    6    2    2    3    2    6    3
## 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126
##    6    2    2    6    2    1    6    6    6    2    6    6    6    6    2    1
## 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142
##    2    6    2    6    6    6    1    1    6    1    6    2    2    6    6    1
## 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158
##    6    2    6    1    1    2    6    1    1    6    2    1    1    1    2    3
## 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174
##    4    3    6    6    6    6    3    1    1    1    1    1    6    6    1    1
## 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190
##    1    1    6    1    6    2    1    1    1    1    1    1    2    4    3    3
## 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206
##    1    2    1    6    1    6    4    2    1    2    6    1    1    3    3    2
## 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222
##    6    6    3    3    2    2    4    5    3    2    3    4    4    3    4    3
## 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238
##    2    3    3    4    3    3    4    3    4    4    6    6    6    6    3    3
## 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254
##    2    2    2    5    4    4    3    6    2    2    2    3    3    2    6    5
## 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270
##    5    6    3    6    2    5    2    1    3    2    2    5    6    6    3    2
## 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286
##    5    4    2    2    2    6    2    4    6    1    2    6    6    6    3    3
## 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302
##    2    3    3    2    6    1    2    6    1    1    3    3    1    1    1    2
## 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318
##    1    6    1    1    1    6    6    6    6    1    1    6    2    1    1    1
## 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334
##    1    1    1    2    6    1    1    1    2    6    2    2    4    4    2    3
## 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350
```

```
##    4    3    4    6    1    6    1    1    6    2    6    2    2    6    6    2
## 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366
##    3    2    2    2    3    2    2    3    5    6    6    6    2    6    6    2
## 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382
##    6    1    2    2    2    6    6    1    1    1    2    1    1    6    1    1
## 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398
##    1    1    2    6    2    6    2    6    6    6    6    6    2    6    6    2
## 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414
##    4    2    2    2    2    3    6    6    2    1    2    6    2    6    3    6
## 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430
##    2    1    1    6    6    6    3    1    2    6    2    6    2    2    2    2
## 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446
##    2    6    6    6    6    1    2    6    6    6    1    6    2    1    6    6
## 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462
##    6    2    6    6    1    6    2    2    6    6    2    3    4    2    2    6
## 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478
##    3    6    6    1    1    2    2    6    6    4    2    6    1    6    2    1
## 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494
##    3    1    2    3    6    2    6    6    2    4    4    3    6    4    4    5
## 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510
##    4    2    6    3    2    4    3    4    4    5    3    2    4    2    3    3
## 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526
##    6    6    2    6    2    2    2    6    6    6    6    2    6    6    6    6
## 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542
##    6    3    2    6    1    1    1    1    2    1    2    6    3    2    6    6
## 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558
##    6    6    6    2    6    1    1    1    1    1    2    6    6    6    1    3
## 1559 1560 1561 1562 1563 1564 1565 1566 1569 1570 1571 1573 1574 1575 1576 1577
##    3    3    3    3    4    4    6    6    3    6    5    2    1    6    6    1
## 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593
##    1    1    6    1    1    1    1    2    2    3    6    6    2    1    3    6
## 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609
##    6    6    1    3    6    5    2    5    5    3    2    2    3    2    3    4
## 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625
##    3    1    6    6    1    3    6    2    2    2    6    6    2    2    1    2
## 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641
##    2    1    1    6    2    6    2    3    3    6    6    4    3    4    4    2
## 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657
##    6    6    2    6    3    3    2    6    2    2    3    2    1    6    1    1
## 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673
##    6    6    6    1    2    3    6    2    1    1    2    6    1    1    1    6
## 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689
##    1    6    6    2    6    1    6    1    6    1    1    2    4    3    2    6
## 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705
##    6    4    6    1    4    2    2    2    3    6    3    2    2    3    4    1
## 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721
##    2    3    2    3    2    2    6    6    2    6    6    6    6    1    1    6
## 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1736 1737 1738 1739 1740
##    1    1    4    4    4    2    2    3    3    6    1    2    6    6    6    6
## 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756
##    6    1    1    1    6    2    2    6    3    3    6    6    6    2    6    2
## 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772
##    6    1    1    1    6    6    6    6    6    1    1    1    6    6    2    6
## 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788
```

```
##    6    2    3    2    6    1    6    2    1    1    6    1    1    1    1    1
## 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804
##    1    6    6    2    6    3    2    2    2    1    2    4    1    3    3    3
## 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820
##    2    2    3    3    2    2    6    6    6    5    2    6    1    2    2    2
## 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836
##    6    6    2    6    2    3    6    6    2    6    6    6    6    6    2    2
## 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852
##    2    6    1    1    6    2    6    6    2    2    2    2    6    2    2    4
## 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868
##    2    3    3    3    4    6    6    6    1    1    1    3    3    2    2    5
## 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884
##    6    1    1    6    1    6    1    1    1    1    6    6    6    2    4    3
## 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900
##    3    2    1    6    1    3    6    2    1    1    1    1    2    6    2    2
## 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916
##    6    6    3    3    6    6    6    3    2    1    2    6    6    1    1    3
## 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932
##    4    1    1    2    1    3    6    6    1    3    1    1    1    3    2    2
## 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948
##    1    6    2    2    6    2    3    3    6    5    4    6    1    1    1    6
## 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964
##    6    1    1    6    1    1    1    6    3    3    2    1    1    1    1    5
## 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980
##    2    4    6    2    2    6    1    2    6    2    2    6    1    2    4    4
## 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996
##    2    2    4    3    3    6    2    6    6    2    2    2    6    6    6    1
## 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
##    6    1    2    6    6    2    6    6    2    6    2    1    6    6    6    1
## 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028
##    2    3    6    6    6    6    2    6    6    6    6    1    6    6    6    6
## 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044
##    1    6    3    6    2    2    6    2    2    6    3    6    2    6    6    2
## 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060
##    2    6    2    6    2    2    2    1    2    2    1    1    2    2    1    6
## 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076
##    2    6    2    2    1    1    2    2    6    2    2    1    1    2    6    1
## 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092
##    2    6    1    2    3    3    1    2    1    6    6    2    6    3    6    6
## 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108
##    1    6    1    3    3    6    1    2    3    2    2    6    6    3    3    2
## 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124
##    3    3    3    2    3    3    6    6    6    3    4    2    2    1    6    6
## 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140
##    1    6    6    6    6    6    2    2    6    3    2    5    2    4    4    6
## 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156
##    1    1    2    6    2    6    2    2    2    2    3    6    6    1    6    6
## 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172
##    1    3    4    4    3    2    6    6    2    2    6    1    6    2    2    6
## 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188
##    6    2    6    6    6    1    6    1    6    6    3    1    6    1    6    6
## 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204
##    6    1    6    2    2    3    2    1    6    3    2    1    1    1    1    1
## 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220
```

```
##    2    1    6    1    6    2    6    6    2    1    5    1    6    6    2    6
## 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236
##    2    1    1    6    6    1    1    6    2    2    6    6    6    6    2    2
## 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252
##    2    2    6    3    2    2    2    1    2    3    2    2    2    1    6    6
## 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268
##    6    2    3    6    2    3    6    2    2    6    6    2    2    4    6    1
## 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284
##    1    1    1    1    4    2    3    1    6    2    6    2    2    2    5    2
## 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300
##    2    2    2    2    6    3    1    3    2    3    2    3    3    3    6    3
## 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316
##    3    2    1    2    6    1    1    1    1    1    6    1    6    3    6    6
## 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332
##    2    2    6    6    1    2    6    1    1    6    3    3    2    3    5    4
## 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348
##    2    3    4    3    4    3    4    4    6    6    6    6    6    6    6    1
## 2349 2350 2351 2352 2353 2354 2355 2357 2358 2359 2360 2361 2362 2363 2364 2365
##    6    1    6    6    2    2    6    2    6    2    2    2    3    2    2    3
## 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381
##    2    3    2    3    2    3    2    2    6    6    4    1    1    6    3    5
## 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397
##    3    6    2    5    5    5    3    3    2    2    4    3    3    6    3    3
## 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413
##    3    4    4    6    6    6    6    6    1    1    6    2    6    2    6    6
## 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429
##    6    1    1    2    2    2    2    4    4    3    4    6    2    2    3    4
## 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445
##    6    2    6    1    3    6    2    2    1    1    6    6    2    6    6    6
## 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461
##    6    6    5    6    2    4    3    3    4    5    2    2    2    6    3    2
## 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477
##    1    6    1    6    4    5    6    2    6    2    5    5    6    5    3    4
## 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493
##    6    2    2    1    6    6    6    3    3    4    4    3    3    3    2    2
## 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509
##    2    6    2    6    2    2    6    2    2    1    6    4    4    3    4    4
## 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525
##    3    2    2    2    4    6    2    1    6    1    3    2    4    1    1    1
## 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541
##    1    1    1    6    6    3    2    2    2    2    3    3    3    6    2    1
## 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557
##    2    2    1    2    6    6    2    3    6    6    6    6    6    6    2    2
## 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573
##    6    2    6    6    2    6    6    2    6    6    6    2    6    6    1    1
## 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589
##    2    2    2    2    2    1    1    6    6    3    6    3    3    6    3    5
## 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605
##    4    2    2    6    6    6    6    6    6    3    2    2    2    3    3    3
## 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621
##    3    3    2    2    3    2    4    4    2    2    3    3    2    3    6    2
## 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637
##    3    6    2    6    6    2    1    1    2    6    6    2    6    2    6    3
## 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653
```

```
##    6    6    6    3    2    3    2    2    6    2    3    3    3    2    2    2
## 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669
##    6    2    6    2    6    1    6    2    6    6    2    6    6    1    1    6
## 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685
##    6    2    6    1    6    1    6    6    2    2    2    6    6    2    4    4
## 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701
##    4    6    2    3    2    2    1    5    5    3    2    3    3    3    3    2
## 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717
##    6    2    3    2    2    2    3    4    3    6    3    3    3    3    3    2
## 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733
##    6    1    4    3    2    6    3    1    6    2    3    2    3    6    3    3
## 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749
##    1    6    6    4    4    4    3    6    1    2    1    1    6    1    2    1
## 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765
##    1    6    1    1    4    6    1    1    4    4    2    2    6    1    3    3
## 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781
##    1    1    1    2    6    6    2    6    1    6    1    1    1    2    6    4
## 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797
##    2    3    2    2    3    2    2    2    2    3    2    2    3    3    4    3
## 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813
##    4    4    3    3    2    2    6    3    2    2    6    2    6    6    1    2
## 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829
##    1    1    2    3    2    2    2    3    2    2    2    2    4    1    2    6
## 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845
##    2    2    1    5    4    3    6    6    2    6    2    2    6    6    6    6
## 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861
##    2    2    3    3    3    3    3    2    4    2    3    2    5    3    3    1
## 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877
##    1    1    1    1    1    1    1    1    1    1    1    1    1    6    3    6
## 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893
##    6    1    6    3    3    3    2    6    1    2    1    3    2    2    6    6
## 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909
##    6    6    2    6    1    5    3    4    4    3    1    1    1    2    2    6
## 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925
##    3    1    6    6    2    6    1    1    1    1    1    2    3    1    1    5
## 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941
##    6    2    6    2    3    4    2    5    4    4    4    4    4    4    5    2
## 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957
##    6    2    6    6    1    6    6    6    6    6    6    6    6    1    6    2
## 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973
##    2    2    2    3    2    3    6    2    3    1    6    6    6    6    2    6
## 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989
##    2    6    1    1    1    4    1    4    3    3    3    3    2    3    3    3
## 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005
##    3    6    3    2    3    3    3    1    1    1    1    1    1    1    6    1
## 3006 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022
##    1    2    1    6    2    2    3    1    1    2    6    1    6    1    2    6
## 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038
##    2    3    1    1    1    1    1    6    6    3    6    6    6    3    2    2
## 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054
##    1    1    1    6    6    6    2    2    3    6    1    3    2    2    2    6
## 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070
##    2    2    2    2    6    2    3    2    2    3    6    2    2    2    6    2
## 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086
```

```
##    1    2    6    2    6    6    6    6    6    6    6    6    6    2    6    3
## 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102
##    2    2    6    2    6    6    6    6    2    6    3    3    2    2    6    6
## 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118
##    2    6    6    6    6    1    1    6    1    3    2    4    6    6    3    2
## 3119 3120 3121 3122 3123
##    2    6    1    6    6
##
## Within cluster sum of squares by cluster:
## [1] 39879927611 69844625647 71441950782 93064476093 61733408117 53101125057
##  (between_SS / total_SS =  89.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Find the mean population, household income and per capita income grouped by clusters. In addition, display the most frequent county that can be observed within each cluster.

```
# creating county_name variable in acs_il_t
acs_il_t$subregion <- gsub("Census Tract ", "", acs_il_t$NAME)
acs_il_t$subregion <- gsub("^[^,]*, ", "", acs_il_t$subregion)


# adding cluster variable
acs_il_t_cluster <- acs_il_t %>%
  mutate(cluster = km$cluster)


# mean population, household income and per capita income grouped by clusters
stat <- acs_il_t_cluster %>%
  group_by(cluster) %>%
  summarize(mean_pop = mean(pop),
            mean_hh_income = mean(hh_income),
            mean_income = mean(income),
            most_frequent_county = names(which.max(table(subregion))))

# table
kable(stat, caption = "Statistics")
```

Table 1: Statistics

cluster	mean_pop	mean_hh_income	mean_income	most_frequent_county
1	3023.996	27265.57	15527.79	Cook County, Illinois
2	4357.746	64282.81	31434.40	Cook County, Illinois
3	5107.751	87549.29	41192.31	Cook County, Illinois
4	4535.430	112380.92	59990.76	Cook County, Illinois
5	4126.795	163369.92	78493.00	Cook County, Illinois
6	4029.991	46117.11	23254.56	Cook County, Illinois

As you might have seen earlier, it's not always clear which number of clusters is the optimal choice. To

automate K Means clustering, program a function based on `kmeans()` that takes K as an argument. You can fix the other arguments, e.g. such that a specific dataset is always used when calling the function.

We want to utilize this function to iterate over multiple Ks (e.g., K = 2, ..., 10) and -- each time -- add the resulting cluster membership as a new variable to our (cleaned) original data frame (`acs_il_t`). There are multiple solutions for this task, e.g. think about the `apply` family or `for` loops.

```
# Specifying the dataset we want to use when calling the function
cluster_data <- acs_il_t %>%
    select(pop, hh_income, income)

# Writing the function to automate K Means clustering
function_kmeans <- function(data, K_clusters) {
  # Looping over each value in K, which could be a vector
  for (k in K_clusters) {
    # Performing K means clustering
  kmeans_result <- kmeans(cluster_data, centers = k, nstart = 10)
  # Adding resulting cluster membership to the original data frame
    cluster_col_name <- paste0("Cluster_K", k)
    data[[cluster_col_name]] <- kmeans_result$cluster
  }
  return(data)
}
```

Finally, display the first rows of the updated data set (with multiple cluster columns).

```
# K values
Ks <- 2:10
head(function_kmeans(acs_il_t, Ks))
```

```
##   state county  tract                                          NAME  pop
## 1    17    031 806002 Census Tract 8060.02, Cook County, Illinois 7304
## 2    17    031 806003 Census Tract 8060.03, Cook County, Illinois 7577
## 3    17    031 806400    Census Tract 8064, Cook County, Illinois 2684
## 4    17    031 806501 Census Tract 8065.01, Cook County, Illinois 2590
## 5    17    031 750600    Census Tract 7506, Cook County, Illinois 3594
## 6    17    031 310200    Census Tract 3102, Cook County, Illinois 1521
##   hh_income income             subregion Cluster_K2 Cluster_K3 Cluster_K4
## 1     56975  23750 Cook County, Illinois          2          2          1
## 2     53769  25016 Cook County, Illinois          2          2          1
## 3     62750  30154 Cook County, Illinois          2          3          1
## 4     53583  20282 Cook County, Illinois          2          2          1
## 5     40125  18347 Cook County, Illinois          2          2          2
## 6     63250  31403 Cook County, Illinois          2          3          1
##   Cluster_K5 Cluster_K6 Cluster_K7 Cluster_K8 Cluster_K9 Cluster_K10
## 1          3          4          2          3          5           7
## 2          3          6          5          3          1           4
## 3          3          4          2          3          5           7
## 4          3          6          5          3          1           4
## 5          4          6          5          2          3          10
## 6          3          4          2          3          5           7
```