多元-05-2020270026

2020270026 王姿文

3/31/2021

1.

$$\sum_{i=1}^{n}(x_i-\bar{x})^2$$
$$=\sum_{i=1}^{n}(x_i-\frac{1}{n}\sum_{j=1}^{n}x_j)^2$$
$$=\sum_{i=1}^{n}(x_i^2-\frac{2}{n}x_i\sum_{j=1}^{n}x_j+(\sum_{j=1}^{n}x_j)^2\frac{1}{n^2})$$
$$=\sum_{i=1}^{n}x_i^2-\frac{2}{n}\sum_{j=1}^{n}x_j\sum_{i=1}^{n}x_i+\frac{1}{n}(\sum_{j=1}^{n}x_j)^2$$
$$=\sum_{i=1}^{n}x_i^2-\frac{1}{n}(\sum_{j=1}^{n}x_j)^2$$
$$=\frac{1}{2n}(n\sum_{i=1}^{n}x_i^2+n\sum_{j=1}^{n}x_j^2-2(\sum_{j=1}^{n}x_j)^2)$$
$$=\frac{1}{2n}(\sum_{i=1}^{n}(nx_i^2+\sum_{j=1}^{n}x_j^2-2\sum_{j=1}^{n}x_jx_i)$$
$$=\frac{1}{2n}(\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^2+x_j^2-2x_ix_j))$$
$$=\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i-x_j)^2$$

2.

2.1 数据

数据为城镇居民消费指标数据，分别为31个省在2019和2011的数据，以下将比较两个数据结果。

下表仅列出2019的数据：

```
load('provConsume.RData')
load('Consume2019.RData')
d1 <- provConsume
d2 <- dataConsume
rn <- d2$X
cn <- c('食品','衣著','居住','家庭設備及用品','交通和通信','文教娛樂','醫療保健','其他')
rownames(d1) <- rn
rownames(d2) <- rn
d1 <- d1[,1:8]
d2 <- d2[,2:9]
colnames(d1) <- cn
colnames(d2) <- cn
kbl(d2) %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = F, font_si
ze = 7)
```

	食品	衣著	居住	家庭設備及用品	交通和通信	文教娛樂	醫療保健	其他

北京	8488.5	2229.5	15751.4	2387.3	4979.0	4310.9	3739.7	1151.9
天津	8983.7	1999.5	6946.1	1956.7	4236.4	3584.4	2991.9	1154.9
河北	4675.7	1304.8	4301.6	1170.4	2415.7	1984.1	1699.0	435.8
山西	3997.2	1289.9	3331.6	910.7	1979.7	2136.2	1820.7	396.5
内蒙古	5517.3	1765.4	3943.7	1185.8	3218.4	2407.7	2108.0	597.1
辽宁	5956.6	1586.1	4417.0	1275.3	2848.5	2929.3	2434.2	756.0
吉林	4675.4	1406.8	3351.5	948.3	2518.1	2436.6	2174.0	564.7
黑龙江	4781.1	1437.6	3314.2	844.8	2317.4	2444.9	2457.1	514.4
上海	10952.6	2071.8	15046.4	2122.8	5355.7	5495.1	3204.8	1355.9
江苏	6847.0	1573.4	7247.3	1496.4	3732.2	2946.4	2166.5	688.1
浙江	8928.9	1877.1	8403.2	1715.9	4552.8	3624.0	2122.6	801.3
安徽	6080.8	1300.6	4281.3	1154.3	2286.6	2132.8	1489.9	411.2
福建	8095.6	1319.6	6974.9	1269.7	3019.4	2509.0	1506.8	619.3
江西	5215.2	1077.6	4398.8	1128.6	2104.3	2094.2	1264.5	367.3
山东	5416.8	1443.1	4370.1	1538.9	2991.5	2409.7	1816.5	440.8
河南	4186.8	1226.5	3723.1	1101.5	1976.0	2016.8	1746.1	354.9
湖北	5946.8	1422.4	4769.1	1418.5	2822.2	2459.6	2230.9	497.5
湖南	5771.0	1262.2	4306.1	1226.2	2538.5	3017.4	1961.6	395.8
广东	9369.2	1192.2	7329.1	1560.2	3833.6	3244.4	1770.4	695.5
广西	5031.2	648.0	3493.2	944.1	2384.7	2007.0	1616.0	294.2
海南	7122.3	697.7	4110.4	932.7	2578.2	2413.4	1294.0	406.2
重庆	6666.7	1491.9	3851.2	1392.5	2632.8	2312.2	1925.4	501.3
四川	6466.8	1213.0	3678.8	1201.3	2576.4	1813.5	1934.9	453.7
贵州	4110.2	984.0	2941.7	873.8	2405.6	1865.6	1274.8	324.3
云南	4558.4	822.7	3370.6	926.6	2439.0	1950.0	1401.4	311.2
西藏	4792.5	1446.3	2320.6	847.7	2015.2	690.3	519.2	397.4
陕西	4671.9	1227.5	3625.3	1151.1	2154.8	2243.4	1977.4	413.3
甘肃	4574.0	1125.3	3440.4	945.3	1972.7	1843.5	1619.3	358.6
青海	5130.9	1359.8	3304.0	953.2	2587.6	1731.8	1995.6	481.8
宁夏	4605.2	1476.6	3245.1	1144.5	3018.1	2352.4	1929.3	525.5
新疆	5042.7	1472.1	3270.9	1159.5	2408.1	1876.1	1725.4	441.7

2011的数据结构：

```
str(d1)
```

```
## 'data.frame':    31 obs. of  8 variables:
##  $ 食品           : num  6906 6663 3927 3558 4962 ...
##  $ 衣著           : num  2266 1755 1426 1462 2514 ...
##  $ 居住           : num  1924 1763 1372 1328 1419 ...
##  $ 家庭設備及用品: num  1563 1175 810 833 1163 ...
##  $ 交通和通信     : num  3521 2700 1527 1488 2004 ...
##  $ 文教娛樂       : num  3307 2116 1204 1419 1812 ...
##  $ 醫療保健       : num  1523 1415 956 851 1239 ...
##  $ 其他           : num  975 837 387 415 765 ...
```

2019的数据结构：

```
str(d2)
```

```
## 'data.frame':    31 obs. of  8 variables:
##  $ 食品           : num  8488 8984 4676 3997 5517 ...
##  $ 衣著           : num  2230 2000 1305 1290 1765 ...
##  $ 居住           : num  15751 6946 4302 3332 3944 ...
##  $ 家庭設備及用品: num  2387 1957 1170 911 1186 ...
##  $ 交通和通信     : num  4979 4236 2416 1980 3218 ...
##  $ 文教娛樂       : num  4311 3584 1984 2136 2408 ...
##  $ 醫療保健       : num  3740 2992 1699 1821 2108 ...
##  $ 其他           : num  1152 1155 436 396 597 ...
```

2.2 用快速聚类法分成4类

均设定指定类个数为四个、指定重复二十次随机初值比较，可以看出2011与2019在分成四个类的条件下，其四个类的省份不同，且四个类的省份个数也不同。

2011:

```
k <- 4
res11 <- kmeans(d1, centers=k, nstart=20)
res11
```

```
## K-means clustering with 4 clusters of sizes 4, 13, 11, 3
##
## Cluster means:
##       食品      衣著      居住 家庭設備及用品 交通和通信 文教娛樂   醫療保健
## 1 7587.390 1965.820 1918.150       1467.118   3672.115 3129.315 1215.3050
## 2 4383.881 1512.278 1122.931        770.340   1460.059 1228.612  846.4638
## 3 5255.227 1630.774 1316.993        925.520   1785.396 1541.681  949.9445
## 4 6419.720 1674.000 1537.673       1182.757   2477.300 2230.183 1050.3667
##       其他
## 1 988.7275
## 2 444.0038
## 3 496.3827
## 4 716.9600
##
## Clustering vector:
##   北京   天津   河北   山西 内蒙古   辽宁   吉林 黑龙江   上海   江苏   浙江
##      1      4      2      2      3      3      2      2      1      4      1
##   安徽   福建   江西   山东   河南   湖北   湖南   广东   广西   海南   重庆
##      3      4      2      3      2      3      3      1      3      3      3
##   四川   贵州   云南   西藏   陕西   甘肃   青海   宁夏   新疆
##      3      2      2      2      3      2      2      2      2
##
## Within cluster sum of squares by cluster:
## [1] 4673100 4922082 5409421 1126730
##  (between_SS / total_SS =  81.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

2019:

```
res12 <- kmeans(d2, centers=k, nstart=20)
res12
```

```
## K-means clustering with 4 clusters of sizes 5, 15, 9, 2
##
## Cluster means:
##       食品      衣著       居住  家庭設備及用品  交通和通信  文教娛樂  醫療保健
## 1 8444.880 1592.360  7380.120       1599.780   3874.880 3181.640  2111.64
## 2 4669.893 1220.367  3428.840       1003.340   2313.133 1978.193  1681.32
## 3 6105.011 1353.600  4191.967       1258.389   2721.456 2432.844  1910.60
## 4 9720.550 2150.650 15398.900       2255.050   5167.350 4903.000  3472.25
##        其他
## 1  791.8200
## 2  412.1067
## 3  495.5111
## 4 1253.9000
##
## Clustering vector:
##   北京   天津   河北   山西 内蒙古   辽宁   吉林 黑龙江   上海   江苏   浙江
##      4      1      2      2      3      3      2      2      4      1      1
##   安徽   福建   江西   山东   河南   湖北   湖南   广东   广西   海南   重庆
##      3      1      2      3      2      3      3      1      2      3      3
##   四川   贵州   云南   西藏   陕西   甘肃   青海   宁夏   新疆
##      3      2      2      2      2      2      2      2      2
##
## Within cluster sum of squares by cluster:
## [1]  9873286 12902304  7099341  4267805
##  (between_SS / total_SS =  92.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

下面按类中心的8个指标平均值降序排序为类 号次序，2019和2011的次序明显不同:

2011:

```
xmean1 <- rowMeans(d1)
cl.new1 <- res11$cluster
cl.new1[] <- as.integer(fct_reorder(factor(cl.new1), xmean1, mean, .desc=TRUE))
kbl(as.data.frame(sort(cl.new1))) %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = F, font_si
ze = 7)
```

	sort(cl.new1)
北京	1
上海	1
浙江	1
广东	1
天津	2
江苏	2

福建	2
内蒙古	3
辽宁	3
安徽	3
山东	3
湖北	3
湖南	3
广西	3
海南	3
重庆	3
四川	3
陕西	3
河北	4
山西	4
吉林	4
黑龙江	4
江西	4
河南	4
贵州	4
云南	4
西藏	4
甘肃	4
青海	4
宁夏	4
新疆	4

2019:

```
xmean2 <- rowMeans(d2)
cl.new2 <- res12$cluster
cl.new2[] <- as.integer(fct_reorder(factor(cl.new2), xmean2, mean, .desc=TRUE))
kbl(as.data.frame(sort(cl.new2))) %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = F, font_si
ze = 7)
```

	sort(cl.new2)
北京	1
上海	1
天津	2

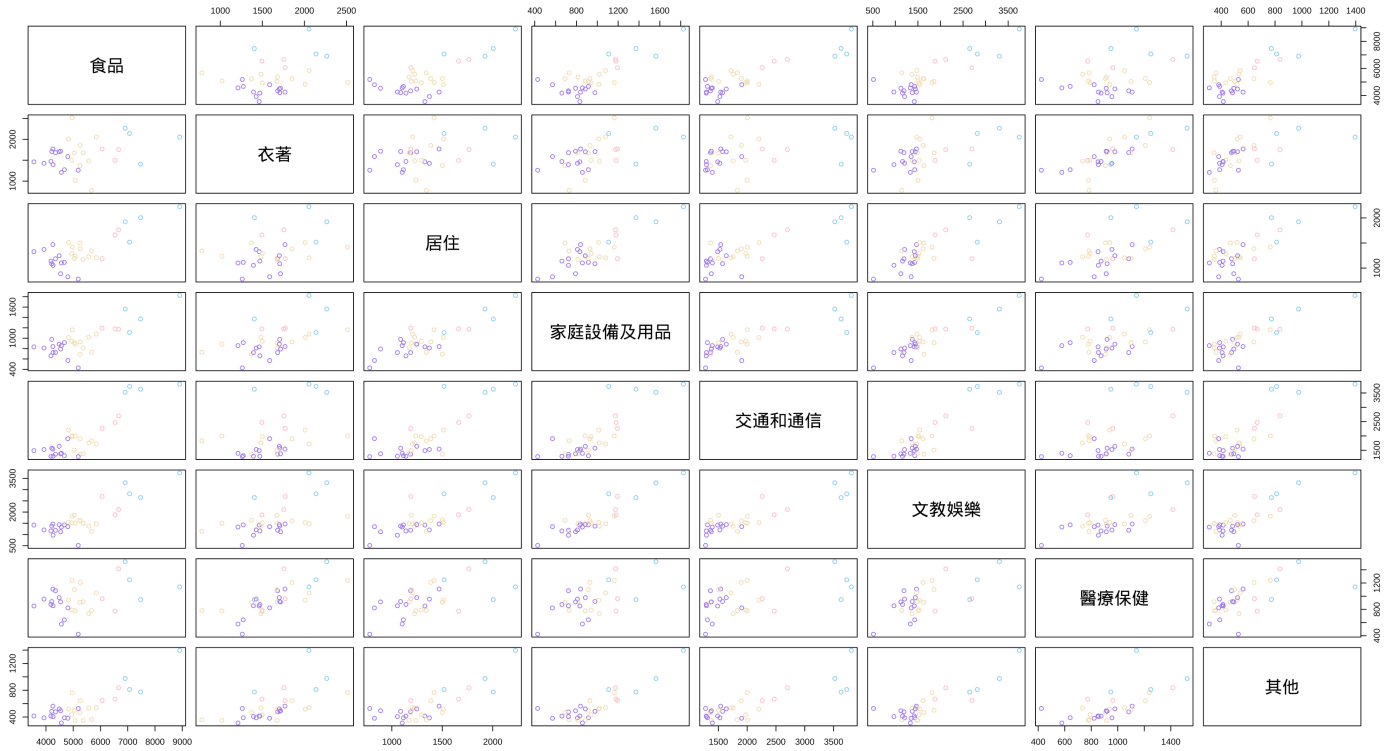
江苏	2
浙江	2
福建	2
广东	2
内蒙古	3
辽宁	3
安徽	3
山东	3
湖北	3
湖南	3
海南	3
重庆	3
四川	3
河北	4
山西	4
吉林	4
黑龙江	4
江西	4
河南	4
广西	4
贵州	4
云南	4
西藏	4
陕西	4
甘肃	4
青海	4
宁夏	4
新疆	4

2.3 用散点图矩阵表现分类效果

下可看出2019和2011的散点图不同，然而由于变量多，所以这样来看散点图的比较还是有些麻烦。

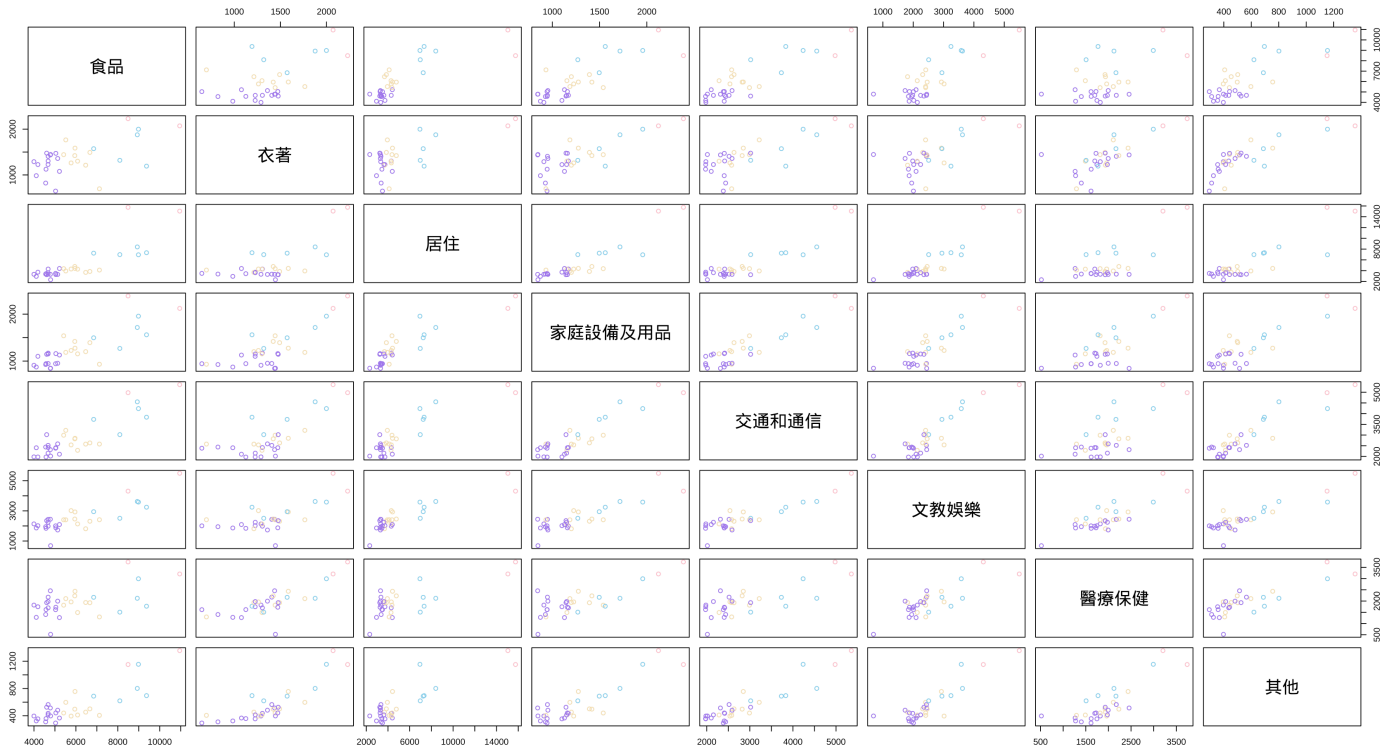
2011:

```
cmap <- c("skyblue", "mediumpurple2", "wheat","pink")
pairs(d1,col=cmap[res11$clust])
```



2019:

```
pairs(d2,col=cmap[res12$clust])
```

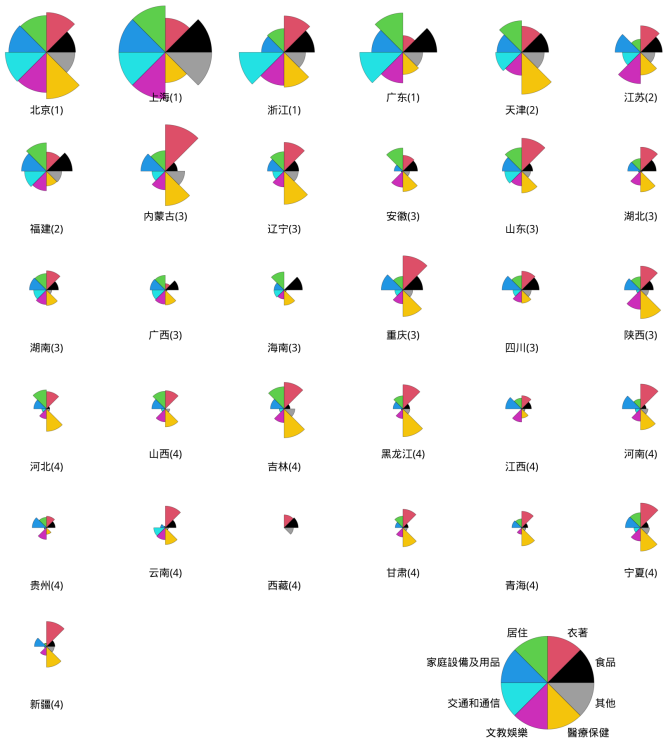


2.4 用星图验证分类

用星图来比较就比散点图还要来得轻易许多，不仅能明确看出个省份所属分类，也能看出其各个变量的占比，可以看出在2011和2019间，不仅分类有变化，变量的占比也有所不同。

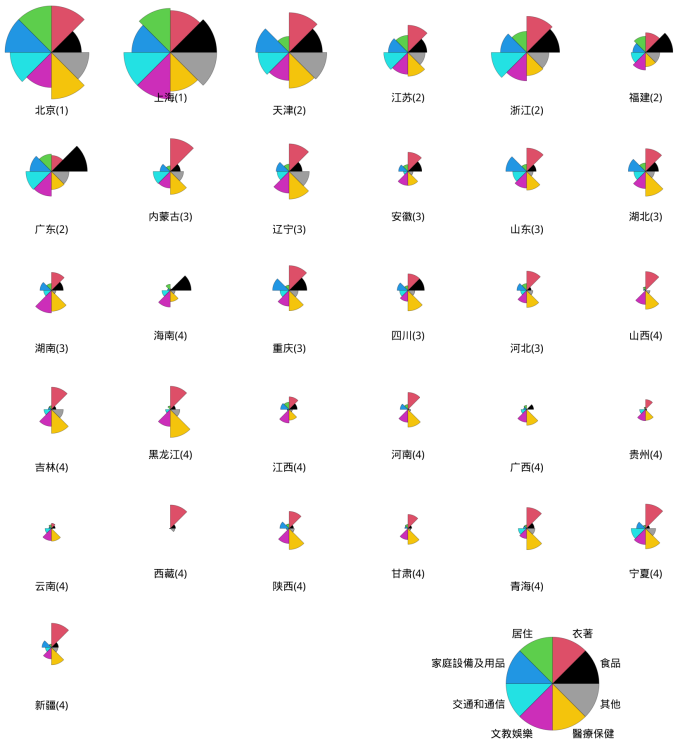
2011:

```
ind1 <- order(cl.new1)
d11 <- d1[ind1,]
names(d11) <- cn
stars(d11, len=0.9, cex=0.8, key.loc=c(12,1.6),draw.segments=TRUE,
      labels = paste(row.names(d11), '(', cl.new1[ind1], ')', sep=''))
```



2019:

```
ind2 <- order(cl.new2)
d12 <- d2[ind2,]
names(d12) <- cn
stars(d12, len=0.9, cex=0.8, key.loc=c(12,1.6), draw.segments=TRUE,
      labels = paste(row.names(d12), '(', cl.new2[ind1], ')', sep=''))
```



2.5 用脸谱图验证分类

接着看脸谱图来验证分类，可以看出2019的分类效果较佳，因为2019的脸谱图在各类比较显然统一。

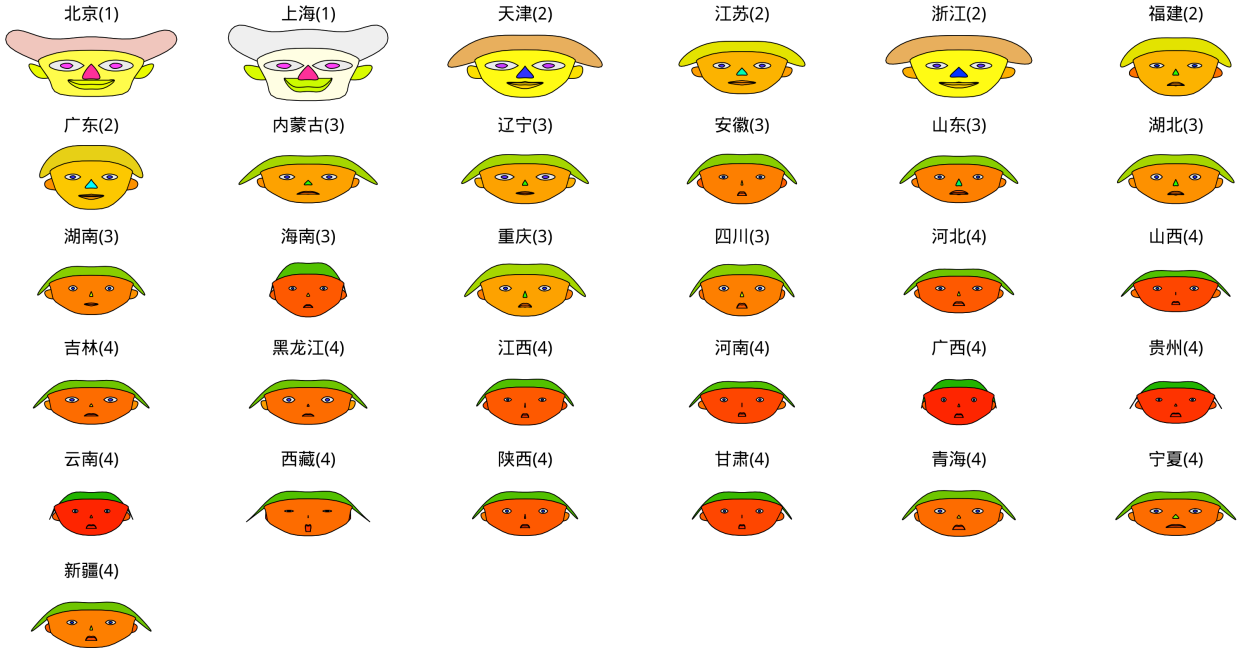
2011:

```
palette(rainbow(12))
aplpack::faces(
  d11, label=paste(row.names(d11), '(', cl.new1[ind1], ')', sep='') )
```

```
## effect of variables:
##  modified item       Var
##  "height of face   " "食品"
##  "width of face    " "衣著"
##  "structure of face" "居住"
##  "height of mouth  " "家庭設備及用品"
##  "width of mouth   " "交通和通信"
##  "smiling          " "文教娛樂"
##  "height of eyes   " "醫療保健"
##  "width of eyes    " "其他"
##  "height of hair   " "食品"
##  "width of hair    "  "衣著"
##  "style of hair    "  "居住"
##  "height of nose   "  "家庭設備及用品"
##  "width of nose    "  "交通和通信"
##  "width of ear     "  "文教娛樂"
##  "height of ear    "  "醫療保健"
```

2019:

```
aplpack::faces(
  d12, label=paste(row.names(d12), '(', cl.new2[ind2], ')', sep='') )
```



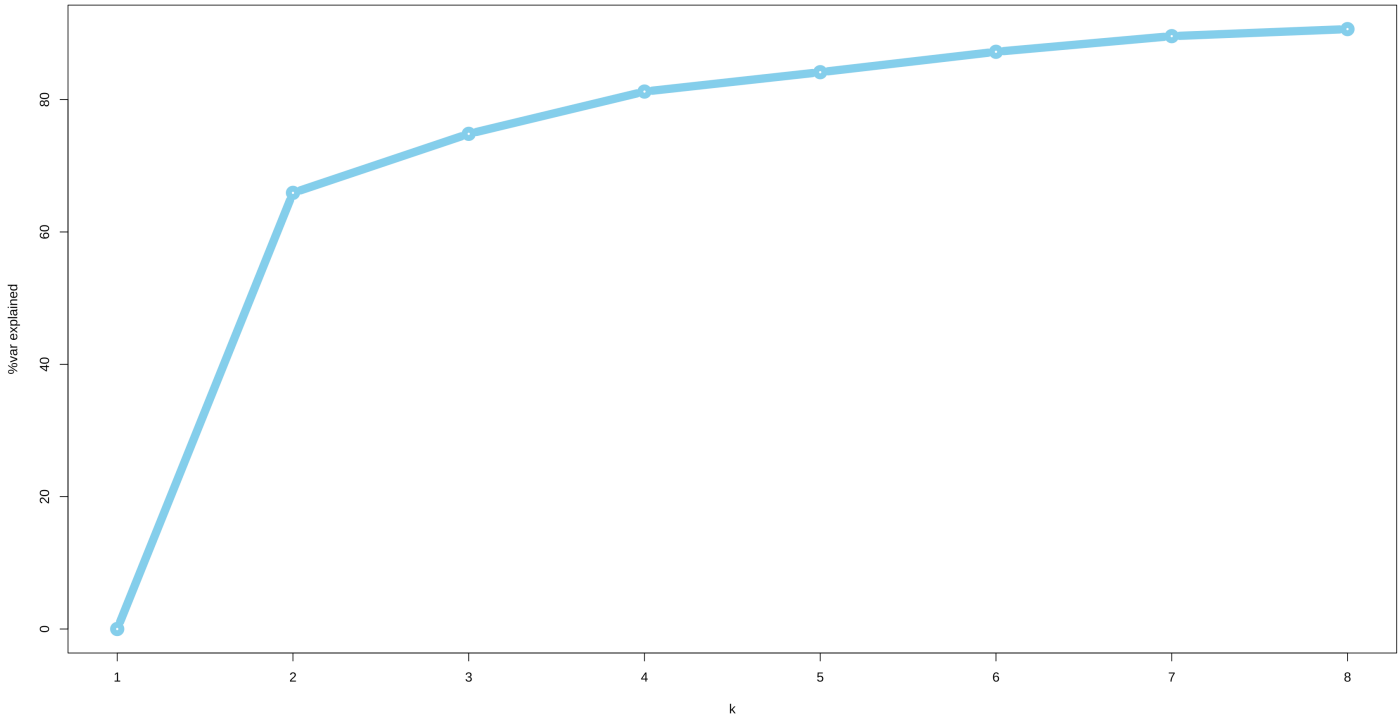
```
## effect of variables:
##  modified item       Var
##  "height of face   " "食品"
##  "width of face    " "衣著"
##  "structure of face" "居住"
##  "height of mouth  " "家庭設備及用品"
##  "width of mouth   " "交通和通信"
##  "smiling          " "文教娛樂"
##  "height of eyes   " "醫療保健"
##  "width of eyes    " "其他"
##  "height of hair   " "食品"
##  "width of hair   "  "衣著"
##  "style of hair   "  "居住"
##  "height of nose  "  "家庭設備及用品"
##  "width of nose   "  "交通和通信"
##  "width of ear    "  "文教娛樂"
##  "height of ear   "  "醫療保健"
```

2.6 分类个数的考察

可以明显看出在分类个数一样的条件下，2019的累积解释变异比2011来得高，因此2019的聚类效果较佳。

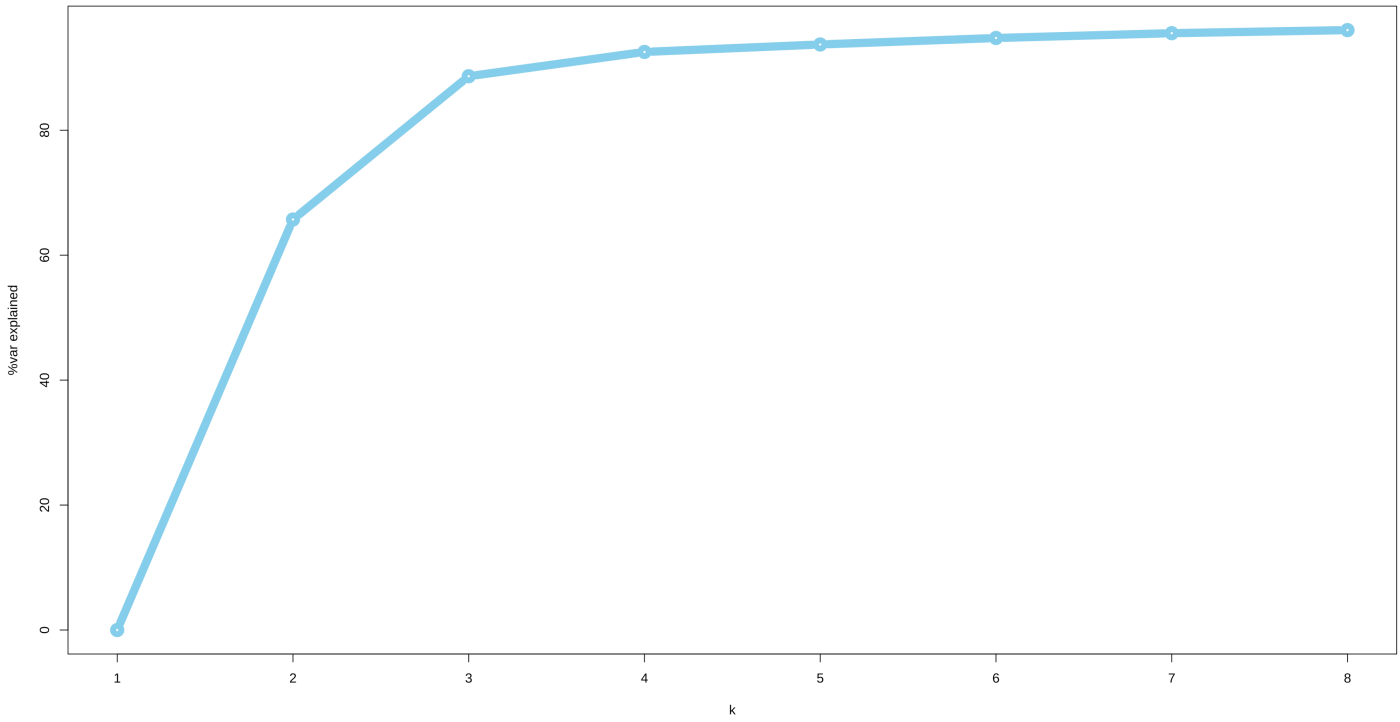
2011:

```
max.k <- 8
rat <- numeric(max.k)
for(kk in seq(max.k)){
  res2 <- kmeans(d1, centers=kk, nstart=20)
  rat[kk] <- (1 - res2$tot.withinss / res2$totss)*100 }
plot(rat, xlab='k', ylab='%var explained', type='b',col = 'skyblue',lwd = 10)
```



2019:

```
max.k <- 8
rat <- numeric(max.k)
for(kk in seq(max.k)){
  res2 <- kmeans(d2, centers=kk, nstart=20)
  rat[kk] <- (1 - res2$tot.withinss / res2$totss)*100 }
plot(rat, xlab='k', ylab='%var explained', type='b',col = 'skyblue',lwd = 10)
```



2.7 小結

整体看来，2019的聚类效果比2011还佳，而经过这8年，聚类的内容也有所改变，2019聚类也和我们所想的较为类似，例如第一类就是两大都市北京和上海。