CS 513 Final - Q3

Aidan Fischer - 10447681

2023-12-18

Load required libraries and data

```
rm(list=ls())
library(caTools)
library(class)
library(e1071)
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
## Warning: package 'lattice' was built under R version 4.3.2
```

```
library(BBmisc)
```

```
## Warning: package 'BBmisc' was built under R version 4.3.2
```

```
##
## Attaching package: 'BBmisc'
```

```
## The following object is masked from 'package:base':
##
##     isFALSE
```

```
data = read.csv("NYNJ_zipcode_population.csv")
data = data[complete.cases(data), ]
```

Take the clustering attributes and normalize

```
datamat = data[,-c(1,2,3)]
rownames(datamat) <- data[,3]
datamat = normalize(datamat, method="scale")
```

Perform kmeans clustering

```
state_kmeans = kmeans(datamat,5)
```

View cluster assignments for cluster 1

```
print(state_kmeans$cluster[state_kmeans$cluster==1])
```

```
##  7086  7094  6390 10018 10038 10044 10307 10464 10471 10501 10502 10505 10506
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10507 10509 10510 10511 10512 10514 10516 10517 10518 10519 10520 10522 10523
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10524 10526 10528 10532 10535 10536 10537 10541 10543 10546 10547 10548 10549
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10552 10553 10560 10567 10570 10576 10577 10578 10579 10580 10588 10590 10591
```

```
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10594 10595 10596 10597 10601 10603 10604 10605 10607 10706 10707 10709 10803
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10805 10901 10913 10914 10915 10916 10917 10918 10919 10920 10921 10922 10923
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10924 10925 10926 10928 10930 10931 10932 10933 10941 10953 10954 10958 10960
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10962 10963 10964 10965 10969 10970 10973 10974 10975 10976 10979 10980 10983
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10984 10985 10986 10987 10988 10989 10990 10992 10993 10994 10996 10998 11004
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11010 11020 11021 11023 11024 11030 11042 11360 11362 11363 11366 11411 11415
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11426 11430 11501 11507 11509 11514 11516 11518 11545 11547 11548 11552 11557
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11558 11559 11560 11563 11565 11568 11569 11576 11577 11579 11581 11596 11598
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11694 11697 11702 11703 11705 11709 11713 11714 11715 11716 11718 11719 11721
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11724 11730 11732 11733 11738 11739 11740 11742 11747 11749 11751 11752 11753
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11754 11755 11762 11764 11766 11767 11768 11769 11770 11771 11776 11777 11778
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11780 11782 11783 11784 11786 11788 11789 11790 11791 11792 11796 11797 11804
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11930 11931 11932 11933 11934 11935 11937 11939 11940 11941 11942 11944 11946
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11947 11948 11949 11952 11953 11954 11955 11956 11957 11958 11959 11960 11961
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11963 11964 11965 11968 11970 11971 11972 11975 11976 11977 11978 11980 12501
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12508 12512 12514 12518 12520 12522 12524 12527 12531 12533 12538 12540 12543
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12545 12546 12549 12553 12563 12564 12566 12567 12569 12570 12571 12572 12574
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12575 12577 12578 12580 12581 12582 12583 12585 12586 12592 12594 12729 12746
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12780
##     1
```

View cluster assignments for cluster 2

```
print(state_kmeans$cluster[state_kmeans$cluster==2])
```

```
##  7047  7087  7093  7305 10002 10027 10029 10031 10032 10033 10451 10452 10453
##     2     2     2     2     2     2     2     2     2     2     2     2     2
## 10456 10457 10458 10459 10460 10462 10463 10466 10467 10468 10469 10472 10473
##     2     2     2     2     2     2     2     2     2     2     2     2     2
## 10701 11203 11204 11206 11207 11208 11210 11212 11213 11214 11218 11219 11220
##     2     2     2     2     2     2     2     2     2     2     2     2     2
## 11221 11223 11225 11226 11229 11230 11233 11234 11235 11236 11354 11355 11368
##     2     2     2     2     2     2     2     2     2     2     2     2     2
## 11372 11373 11377 11385 11419 11432 11434 11435 11550 11706 11717
##     2     2     2     2     2     2     2     2     2     2     2
```

View cluster assignments for cluster 3

```
print(state_kmeans$cluster[state_kmeans$cluster==3])
```

```
##  7002  7030  7302 10001 10003 10009 10011 10013 10016 10019 10021 10023 10024
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 10025 10028 10036 10128 10305 10306 10308 10309 10312 10314 10461 10465 10475
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 10562 10583 10598 10704 10710 10940 10956 11001 11003 11040 11050 11101 11103
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 11105 11106 11201 11209 11211 11215 11216 11217 11222 11231 11238 11357 11364
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 11374 11375 11379 11414 11510 11530 11554 11561 11566 11570 11572 11580 11590
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 11704 11710 11720 11725 11727 11729 11731 11735 11741 11743 11746 11756 11757
##     3     3     3     3     3     3     3     3     3     3     3     3     3
## 11758 11763 11772 11779 11787 11793 11795 11801 11803 12590 12603
##     3     3     3     3     3     3     3     3     3     3     3
```

View cluster assignments for cluster 4

```
print(state_kmeans$cluster[state_kmeans$cluster==4])
```

```
##  7310  7311 10004 10005 10006 10007 10010 10012 10014 10017 10022 10065 10069
##     4     4     4     4     4     4     4     4     4     4     4     4     4
## 10075 10119 10162 10165 10170 10280 10282 10503 10504 10527 10530 10533 10538
##     4     4     4     4     4     4     4     4     4     4     4     4     4
## 10589 10708 10804 10968 11005 11109 11765 11962 11973
##     4     4     4     4     4     4     4     4     4
```

View cluster assignments for cluster 5

```
print(state_kmeans$cluster[state_kmeans$cluster==5])
```

```
##  7029  7032  7304  7306  7307 10026 10030 10034 10035 10037 10039 10040 10301
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 10302 10303 10304 10310 10454 10455 10470 10474 10545 10550 10566 10573 10606
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 10703 10705 10801 10927 10950 10952 10977 11096 11102 11104 11205 11224 11228
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 11232 11237 11239 11356 11358 11361 11365 11367 11369 11370 11378 11412 11413
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 11416 11417 11418 11420 11421 11422 11423 11427 11428 11429 11433 11436 11520
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 11542 11553 11575 11691 11692 11693 11701 11722 11726 11798 11901 11950 11951
##     5     5     5     5     5     5     5     5     5     5     5     5     5
## 11967 12550 12601 12771
##     5     5     5     5
```

KMeans did a much better job at clustering more evenly than single linkage hierarchal clustering did.