CS 513 Final - Q2

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
```

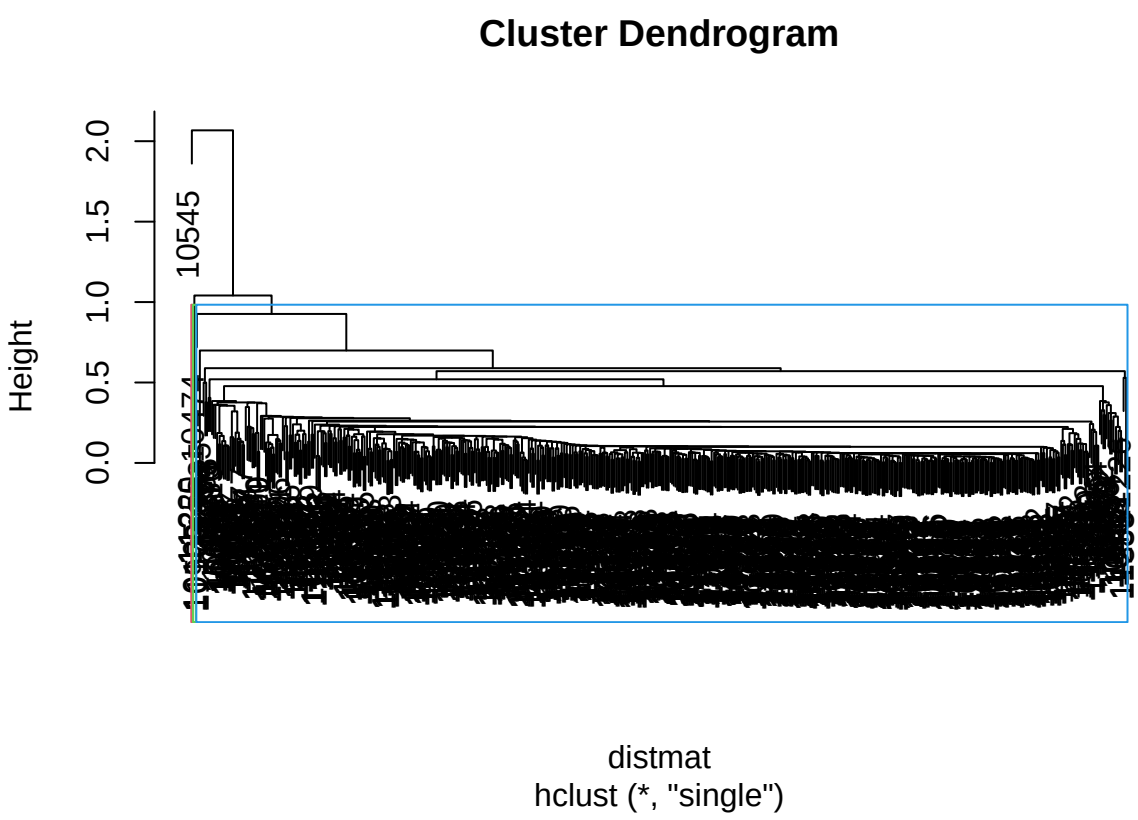
```
data = read.csv("NYNJ_zipcode_population.csv")
data = data[complete.cases(data), ]
```

Take the clustering attributes and normalize

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

Perform hierarchical clustering

```
distmat = stats::dist(datamat, method="euclidean")
state_hclust = hclust(distmat, method="single")
plot(state_hclust)
rect.hclust(state_hclust, k=3, border=2:6)
```




```
clusters = cutree(state_hclust,k=3)
```

View cluster assignments for cluster 1

```
print(clusters[clusters==1])
```

```
##  7002  7029  7030  7032  7047  7086  7087  7093  7094  7302  7304  7305  7306
##     1     1     1     1     1     1     1     1     1     1     1     1     1
##  7307  7310  7311  6390 10001 10002 10003 10004 10005 10006 10007 10009 10010
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10011 10012 10013 10014 10016 10017 10018 10019 10021 10022 10023 10024 10025
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10026 10027 10028 10029 10030 10031 10032 10033 10034 10035 10036 10037 10038
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10039 10040 10044 10065 10069 10075 10119 10128 10162 10165 10170 10280 10282
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10301 10302 10303 10304 10305 10306 10307 10308 10309 10310 10312 10314 10451
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10452 10453 10454 10455 10456 10457 10458 10459 10460 10461 10462 10463 10464
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10465 10466 10467 10468 10469 10470 10471 10472 10473 10474 10475 10501 10502
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10503 10504 10505 10506 10507 10509 10510 10511 10512 10514 10516 10517 10518
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10519 10520 10522 10523 10524 10526 10527 10528 10530 10532 10533 10535 10536
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10537 10538 10541 10543 10546 10547 10548 10549 10550 10552 10553 10560 10562
```

```
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10566 10567 10570 10573 10576 10577 10578 10579 10580 10583 10588 10589 10590
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10591 10594 10595 10596 10597 10598 10601 10603 10604 10605 10606 10607 10701
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10703 10704 10705 10706 10707 10708 10709 10710 10801 10803 10804 10805 10901
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10913 10914 10915 10916 10917 10918 10919 10920 10921 10922 10923 10924 10925
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10926 10927 10928 10930 10931 10932 10933 10940 10941 10950 10952 10953 10954
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10956 10958 10960 10962 10963 10964 10965 10968 10969 10970 10973 10974 10975
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10976 10977 10979 10980 10983 10984 10985 10986 10987 10988 10989 10990 10992
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 10993 10994 10996 10998 11001 11003 11004 11005 11010 11020 11021 11023 11024
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11030 11040 11042 11050 11096 11101 11102 11103 11104 11105 11106 11109 11201
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11216 11217 11218 11219 11221 11222 11223 11224 11225 11226 11228 11229 11230
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11231 11232 11233 11234 11235 11236 11237 11238 11239 11354 11356 11357 11358
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11360 11361 11362 11363 11364 11365 11366 11367 11368 11369 11370 11372 11373
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11374 11375 11377 11378 11379 11385 11411 11412 11413 11414 11415 11416 11417
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11418 11419 11420 11421 11422 11423 11426 11427 11428 11429 11430 11432 11433
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11434 11435 11436 11501 11507 11509 11510 11514 11516 11518 11520 11530 11542
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11545 11547 11548 11550 11552 11553 11554 11557 11558 11559 11560 11561 11563
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11565 11566 11568 11569 11570 11572 11575 11576 11577 11579 11580 11581 11590
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11596 11598 11691 11692 11693 11694 11697 11701 11702 11703 11704 11705 11706
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11709 11710 11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11724
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11725 11726 11727 11729 11730 11731 11732 11733 11735 11738 11739 11740 11741
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11742 11743 11746 11747 11749 11751 11752 11753 11754 11755 11756 11757 11758
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11762 11763 11764 11765 11766 11767 11768 11769 11770 11771 11772 11776 11777
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11778 11779 11780 11782 11783 11784 11786 11787 11788 11789 11790 11791 11792
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11793 11795 11796 11797 11798 11801 11803 11804 11901 11930 11931 11932 11933
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11934 11935 11937 11939 11940 11941 11942 11944 11946 11947 11948 11949 11950
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11951 11952 11953 11954 11955 11956 11957 11958 11959 11960 11961 11962 11963
```

```
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 11964 11965 11967 11968 11970 11971 11972 11973 11975 11976 11977 11978 11980
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12501 12508 12512 12514 12518 12520 12522 12524 12527 12531 12533 12538 12540
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12543 12545 12546 12549 12550 12553 12563 12564 12566 12567 12569 12570 12571
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12572 12574 12575 12577 12578 12580 12581 12582 12583 12585 12586 12590 12592
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 12594 12601 12603 12729 12746 12771 12780
##     1     1     1     1     1     1     1
```

View cluster assignments for cluster 2

```
print(clusters[clusters==2])
```

```
## 10545
##     2
```

View cluster assignments for cluster 3

```
print(clusters[clusters==3])
```

```
## 11220 11355
##     3     3
```

The algorithm seems to assign most of the zip codes to cluster 1 for some reason when using the single method. I was unable to affect this behavior. I am unsure why.