hw 9

2022-10-31

```r
data <- read.csv( file = "social_marketing.csv", row.names = 1)

library(ggplot2)

sum_columns <- colSums(data, na.rm = TRUE)
summary(sum_columns)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      51    5294    7036    9050   11650   34671
```

First I start by looking at the share of data each category of tweets holds by summing the number of tweets per column. I then look at the summary to understand the data more where it can be seen that the maximum number of tweets were in the category "chatter" meaning that a large number of tweets didn't correspond to a category and were vague. The next biggest category was just photo sharing followed by the "health nutrition" category. This shows that a majiority of the followers are focused on health nutrition when they're not talking about random topics or sharing photos.

```r
lm(chatter ~ health_nutrition, data= data)
```

```
##
## Call:
## lm(formula = chatter ~ health_nutrition, data = data)
##
## Coefficients:
##      (Intercept)  health_nutrition
##         4.391245          0.002926
```

```r
lm(chatter ~ cooking, data= data)
```

```
##
## Call:
## lm(formula = chatter ~ cooking, data = data)
##
## Coefficients:
## (Intercept)      cooking
##    4.394477     0.002142
```

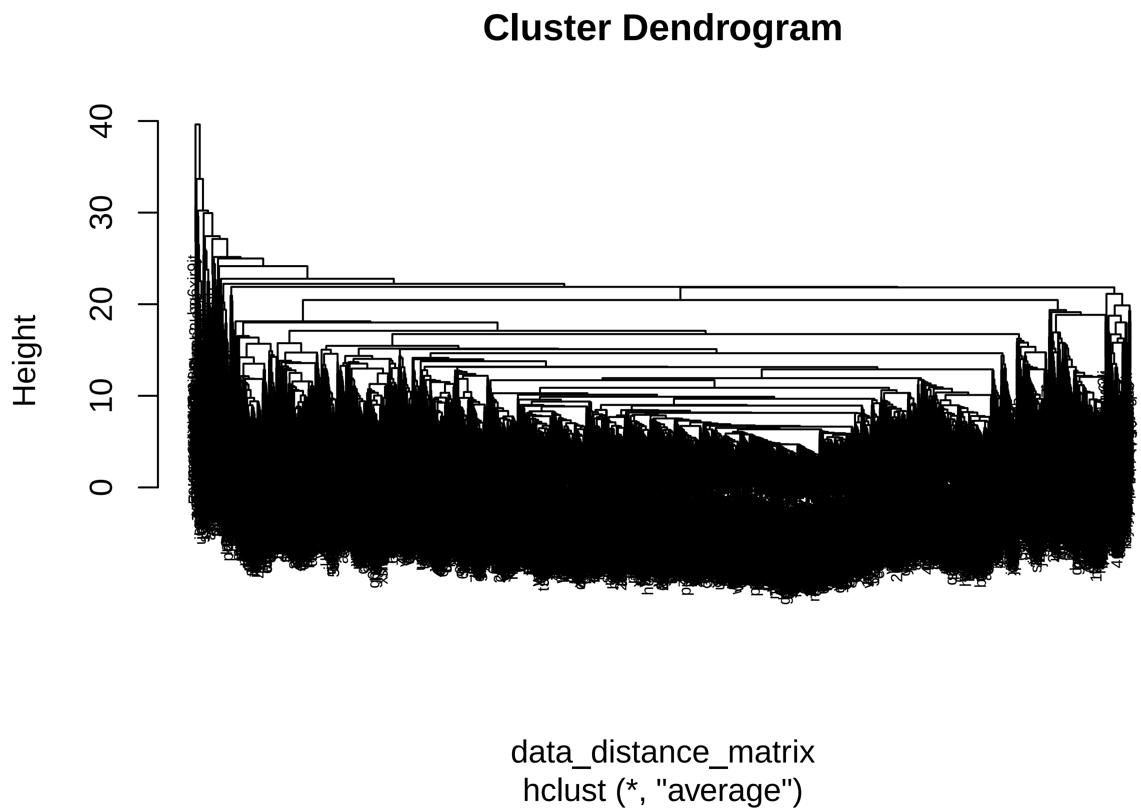
```r
# Here I was trying to figure out whether there is any relation between the top categories and to mainl


lm(health_nutrition ~ cooking + politics + sports_fandom + college_uni , data= data)
```

```
##
## Call:
## lm(formula = health_nutrition ~ cooking + politics + sports_fandom +
##     college_uni, data = data)
##
## Coefficients:
##   (Intercept)        cooking       politics  sports_fandom    college_uni
##       2.07195        0.32784       -0.02069       -0.02347       -0.05511
```

```
#  I also tried looking for relationships between the highest tweeted topics amongst followers of the c
```

I'm now doing hierarchical clustering to see what specific kind of users are clustered together to see if they make a market segment or provide insights into how distributed the followers' thought process is. I'm clustering in all 4 ways to see which method works best.

```
#Average Method

# center/scaling the data
data_scaled <- scale(data, center=TRUE, scale=TRUE)

#Forming a pairwise distance matrix using the dist function

data_distance_matrix = dist(data, method='euclidean')

# running hierarchical clustering
hier_data = hclust(data_distance_matrix, method='average')

plot(hier_data, cex =0.5)
```




```
# Cutting the tree into 10 clusters
cluster1 = cutree(hier_data, k=10)
summary(factor(cluster1))
```

```
##    1    2    3    4    5    6    7    8    9   10
## 7738    1   46   68   13    4    8    2    1    1
```

```
# Examining the cluster members
which(cluster1 == 2)
```

```
## tmf4x9sbh
##        28
```

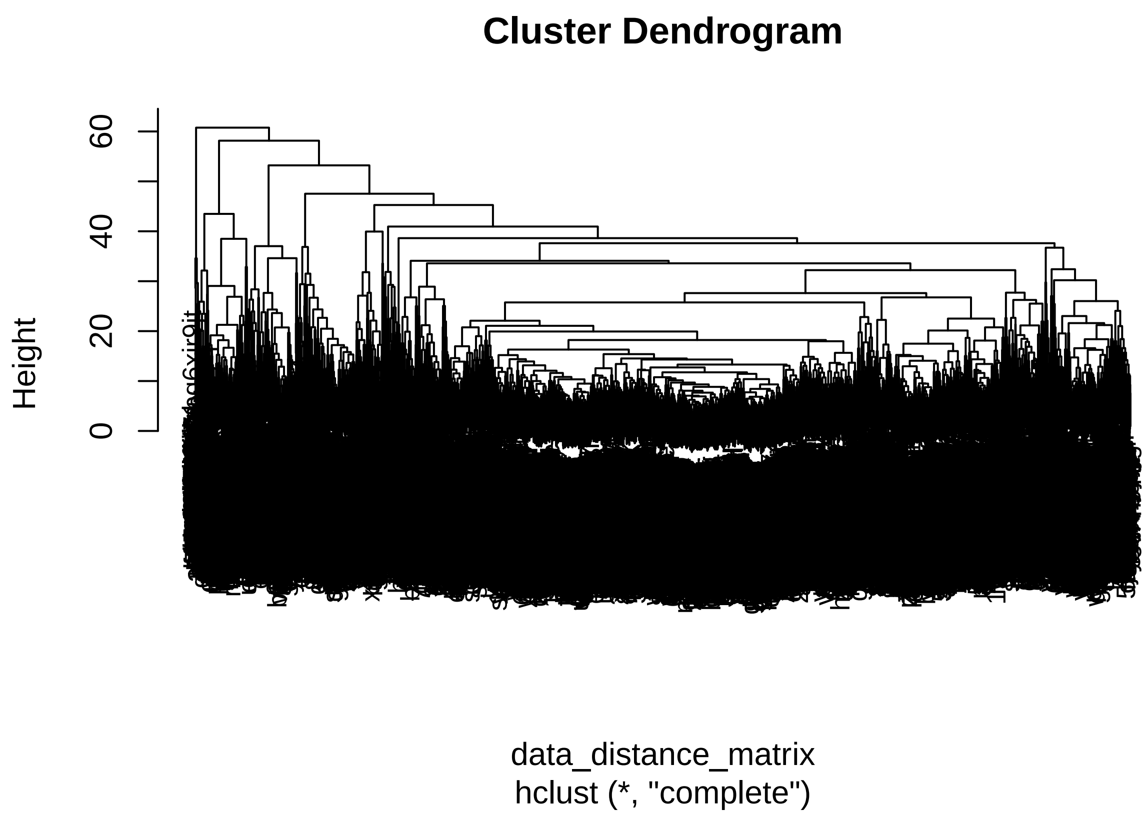
```
which(cluster1 == 6)
```

```
## 9xr7h6jkn wuvbypa9o m6dwl3z9u n5fjuod71
##      2566      2834      4911      5253
```

```
which(cluster1 == 5)
```

```
## lq5aencvm x1m7wfp24 sbm1o46hg txyump81e 8k2fe14zy 5qd9of3mi ywg2rldbe p961kl8vq
##       295       749      1311      1342      2437      2895      4972      5147
## 7vj2b9ogq sbo8lrgy2 hgwblyq4o v1o65a3yl drujonq46
##      5270      6041      6118      6427      7660
```

```r
# After looking at the summary, the clusters are not very evenly split where some only have 1-2 element

# Using max ("complete") linkage
hier_data2 = hclust(data_distance_matrix, method='complete')

# Plot the dendrogram
plot(hier_data2, cex=0.8)
```




```r
cluster2 = cutree(hier_data2, k=10)
summary(factor(cluster2))
```

```
##    1    2    3    4    5    6    7    8    9   10
##  347 6155  489   70  110  420  218   31   22   20
```

```r
# Examine the cluster members
which(cluster2 == 8)
```

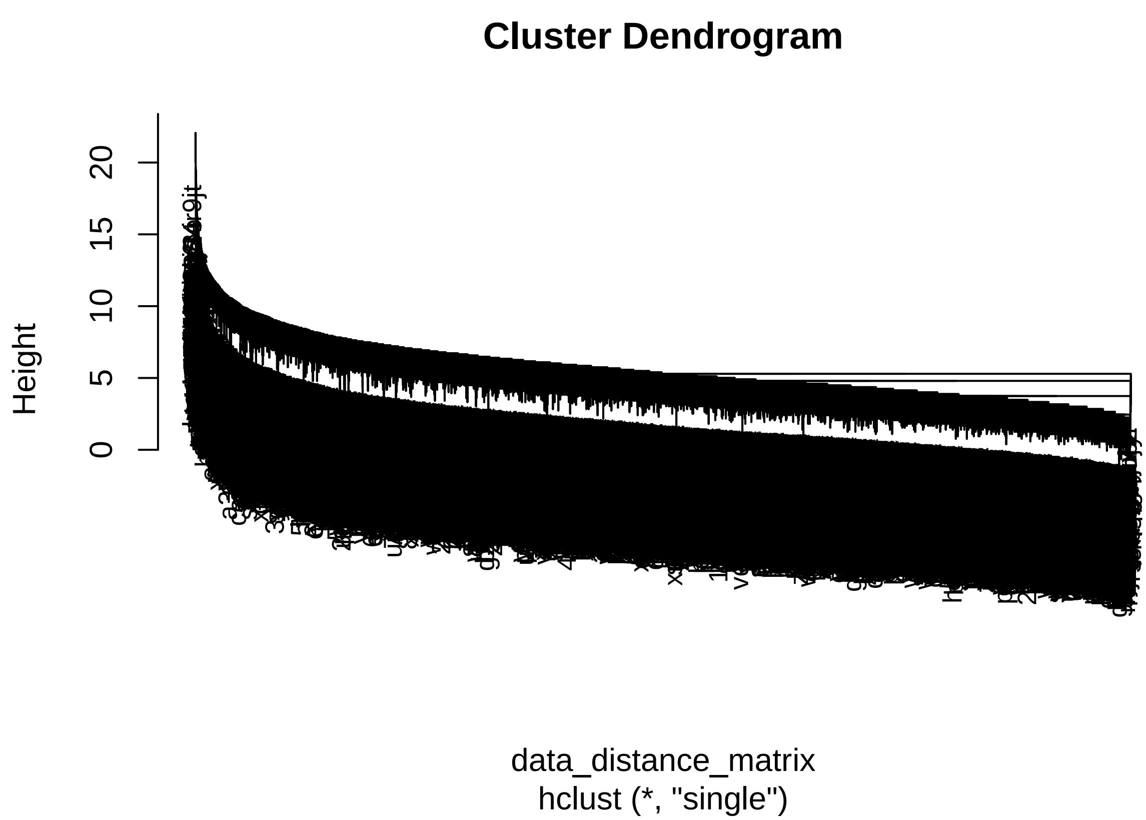
```
## 59ba8dqsp 68ylut5dp 36jecsyr9 kjsf5ez64 yrustim8o tc12ejw5o 9xr7h6jkn coqmjs5zg
##       506       515      1220      1280      1725      1804      2566      2596
## laugxoe4c k4f6iy9wz vkp4rc29m 7tx3gyj9e fos41vqj7 9ecu45n76 sqzlocjip bq6xir9jt
##      2689      3139      3169      3294      3546      4223      4230      4556
## shx2ikoj1 9qky82o7f m6dwl3z9u xfkp4ib7z l2kntf1ij n5fjuod71 c4n9tb36p espbthq9l
##      4776      4821      4911      4957      5169      5253      5462      5974
```

```
## vljcrs37m htrbuvq5c oru5dmqc8 eptml3fvu sfwvbzp9a awi51f3be 54pe8ywgz
##      6448      6493      6494      6688      6696      7398      7825
```

```
which(cluster2 == 9)
```

```
## ecyfuwq27 h2vs9npt4 tzkcngx7v he58ip2sx 7jy3iarfd wuvbypa9o lmawt78zb ueyjs9th6
##       583       914      1017      1448      2226      2834      3562      3604
## vqtygduaf f48p3owtn jtvwihx8b a6vdqbnz4 f2e7xnwq8 3f6dbz2oe luj5mr7ei qek5oljh1
##      4126      4973      5160      5726      5773      6169      6440      7059
## ufkjds67m 6uh1ci7px 2c57b6ezf 2mj68lx4s qznx1fta8 auose94hp
##      7262      7327      7391      7426      7599      7810
```

```
# After looking at the summary, the clusters are better split in the complete method than in average bu


# Using max ("single") linkage - minimum
hier_data3 = hclust(data_distance_matrix, method='single')

# Plot the dendrogram
plot(hier_data3, cex=0.8)
```



```
cluster3 = cutree(hier_data3, k=10)
summary(factor(cluster3))
```

```
##    1    2    3    4    5    6    7    8    9   10
## 7872    1    1    2    1    1    1    1    1    1
```

```r
# Examine the cluster members
which(cluster3 == 3)
```

```
## syjxdeh26
##      3197
```

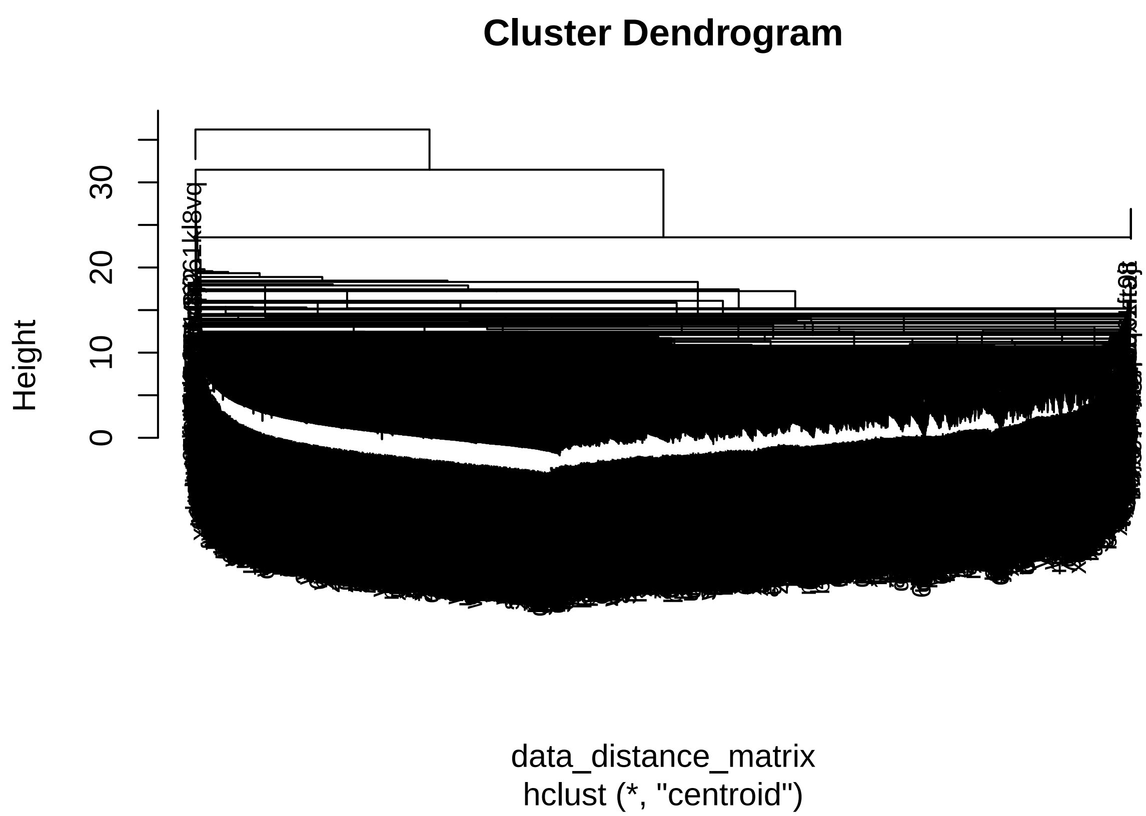
```r
which(cluster3 == 2)
```

```
## p69hzqjo4
##      2260
```

```r
which(cluster3 == 10)
```

```
## qznx1fta8
##      7599
```

```r
# Using centriod linkage
hier_data4 = hclust(data_distance_matrix, method='centroid')

# Plot the dendrogram
plot(hier_data4, cex=0.8)
```



```r
cluster4 = cutree(hier_data4, k=10)
summary(factor(cluster4))
```

```
##    1    2    3    4    5    6    7    8    9   10
## 7872    1    1    1    2    1    1    1    1    1
```

```r
# Examine the cluster members
which(cluster4 == 10)
```

```
## qznx1fta8
##      7599
```

```r
which(cluster4 == 2)
```

```
## mkd5pn8ji
##      3295
```

```r
which(cluster4 == 3)
```

```
## bq6xir9jt
##      4556
```

```r
# both centriod and minimum linkage provided results where clusters has very small elements and usually

# Finally, I conclude using the second clustering method worked best and through which we now have 10 c
```