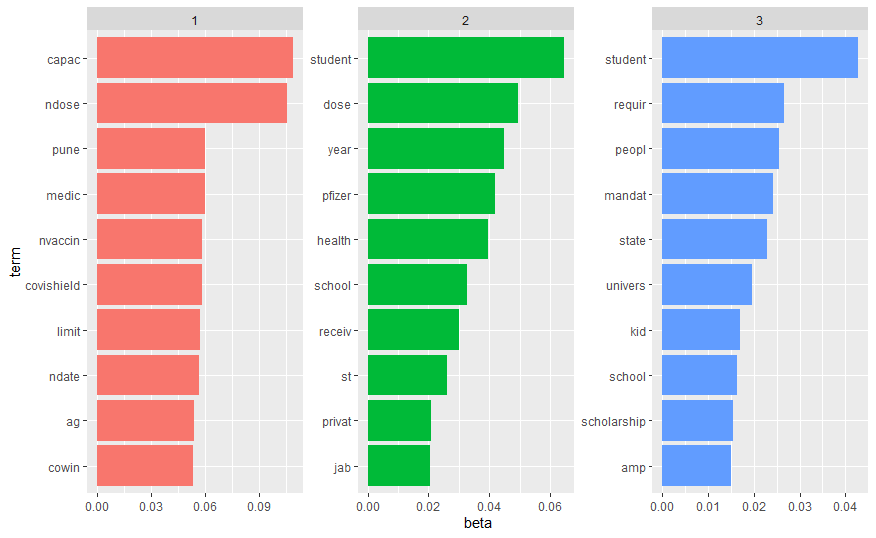
I tried tom implement LDA, Dr Ferg’s, k-means, k-medoids and Monocle3. Based on what I did before Tuesday, I had a feeling that a great proportion of the tweets were the Pin code/Pincode/vaccine slot availability ones. So what I did was I tried clustering on the whole data (after preprocessing etc ) and then I removed the pincode tweets and applied clustering to the remaining tweets.

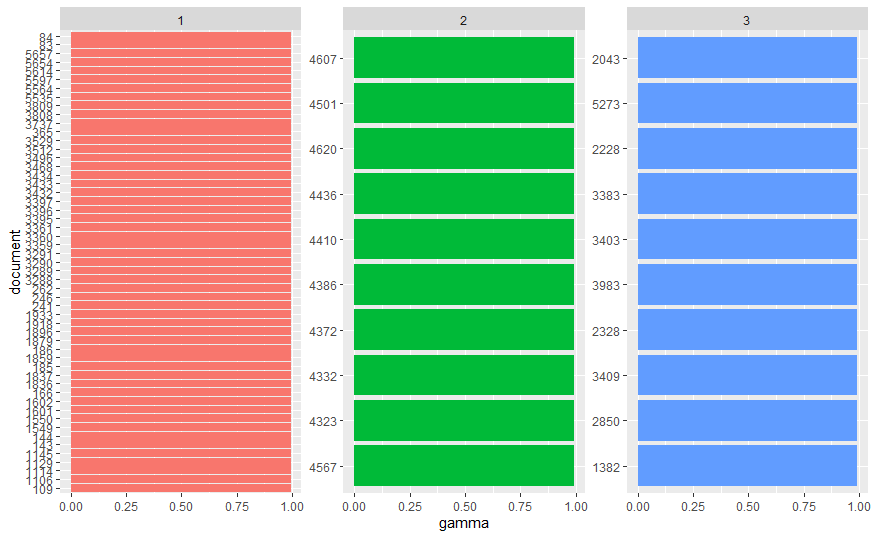
**LDA:**

**Whole data**

I tried clustering into 3 categories (based on similar explanations we arrived at last week). Top 10 most frequently occurring words (based on probability) belonging to each of the 3 topics :

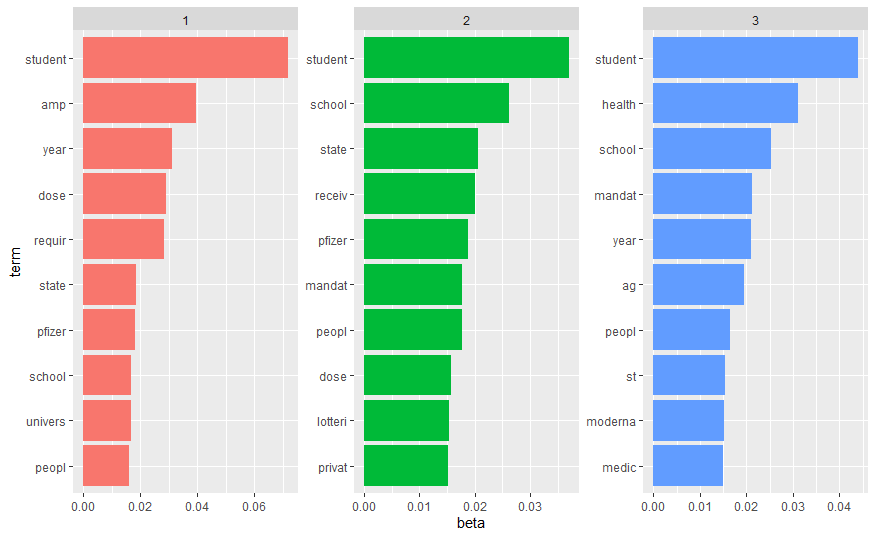
A similar plot with the top 10 or so (?) frequently occurring documents in each topic is on the next page. I checked the tweets with the document numbers in the plots. The subject matter of the documents in each of the 3 categories were more or less similar, and contained the corresponding top 10 words, shown in the plot above. So basically the top 10 frequent words in each category form the same “latent topics”, which are in accordance with the sucject matter of the tweets belonging to each category.

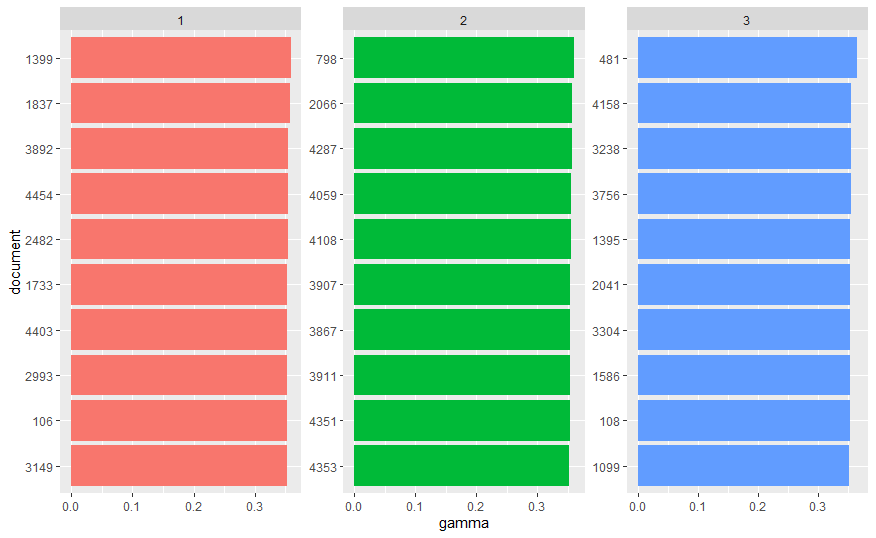
But for subsequent lda clusterings, I didn’t check the tweets belonging to each of the categories anymore. (Sorry!)



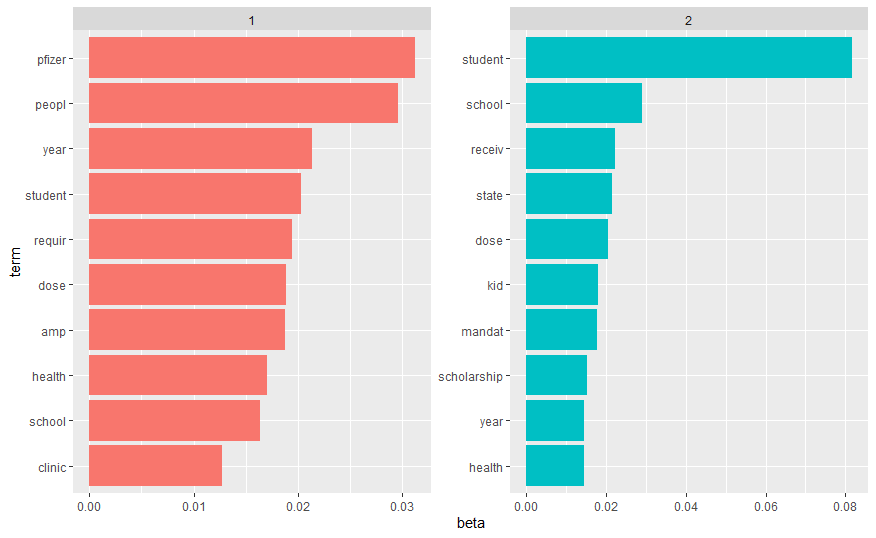
**Restricted Data**

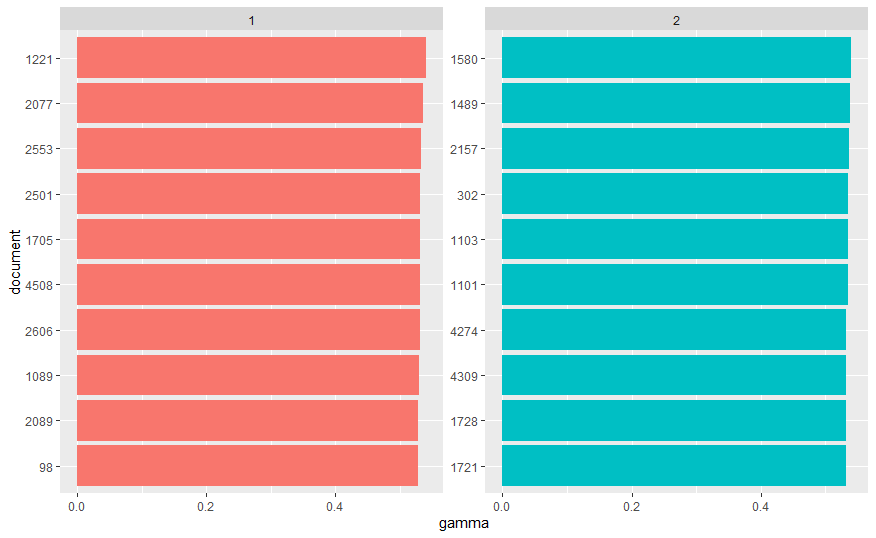
3 categories





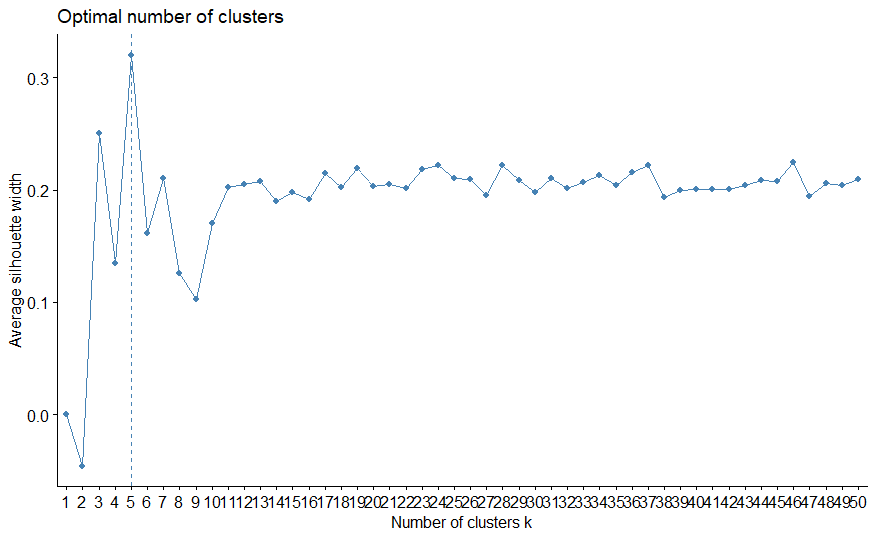
2 categories (since for the whole data, we decided that 1 category should be for informational tweets and a great chunk of these tweets were constituted of the pincode ones, which were removed in this case):

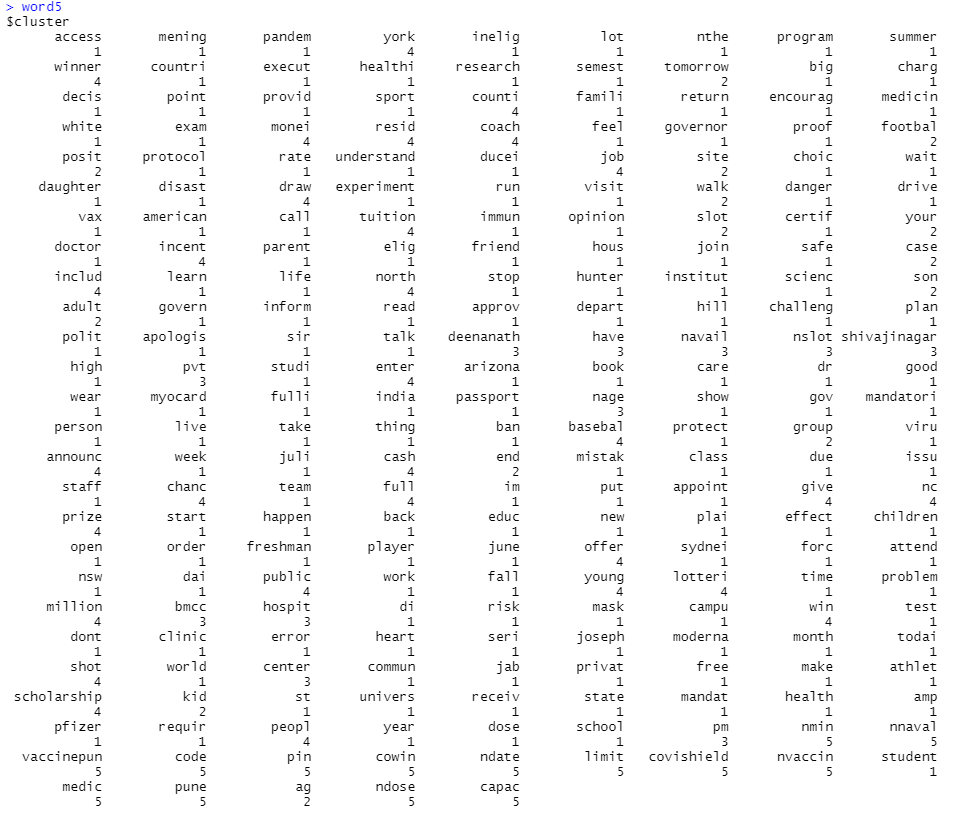




**Dr Ferg’s Method**

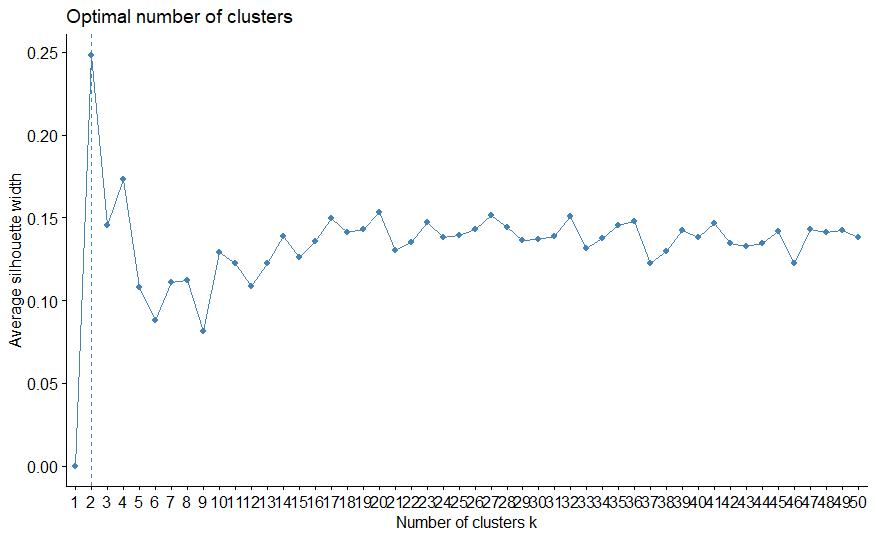
I tried to create the “”word distance matrix” and based on that performed k-medoids classification. But I couldn’t find a way to access the distances each point in each cluster were away from each other and from other points in other clusters. So I couldn’t make a plot of the clusters. **Whole Data:**

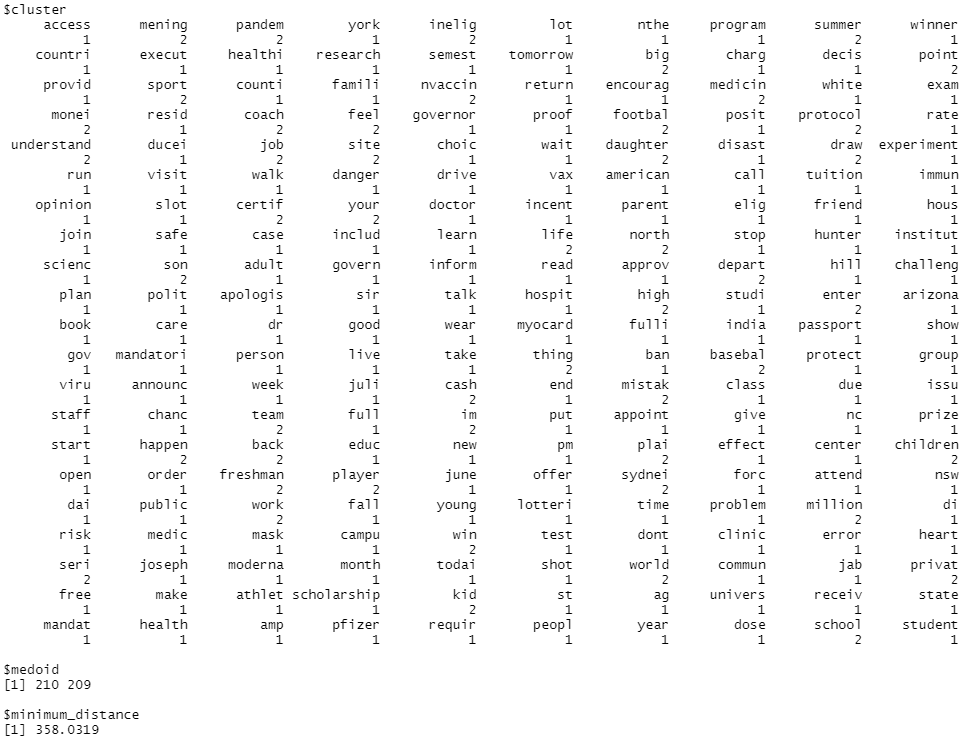


The silhouette method for determining optimal number of clusters. In this case I used k-means algorithm. I think for k-medoids the optimal number of clusters would have increased. But I performed k-medoids with 5 clusters nevertheless.

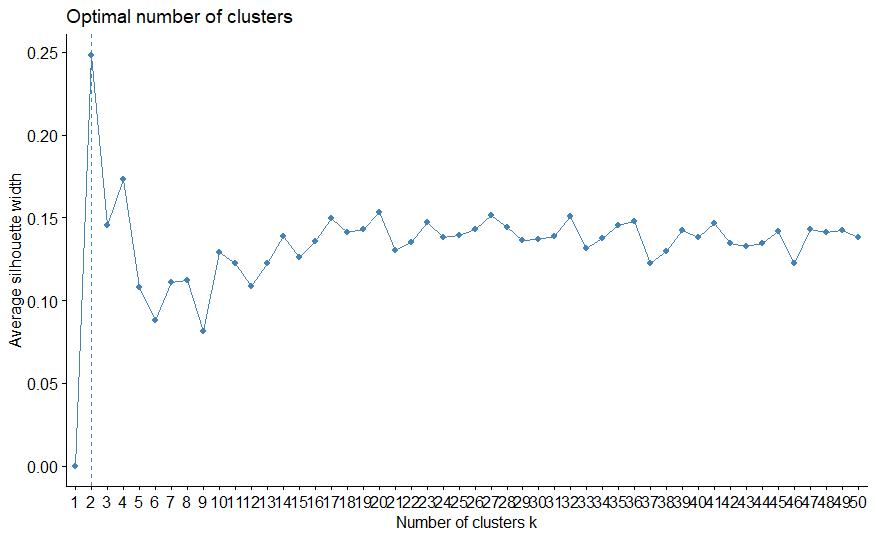
The medoids were 225, 228, 95, 199 and 215 respectively. The value of the category minimum distance in the summary of the classification was 335.2886

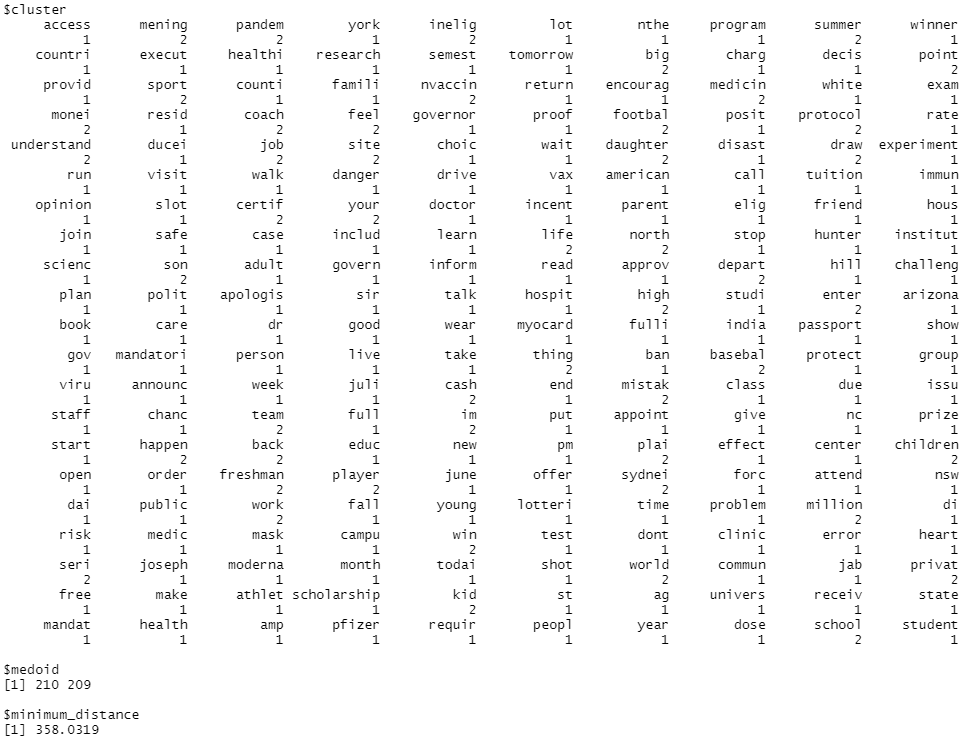
**Restricted Data:**

****



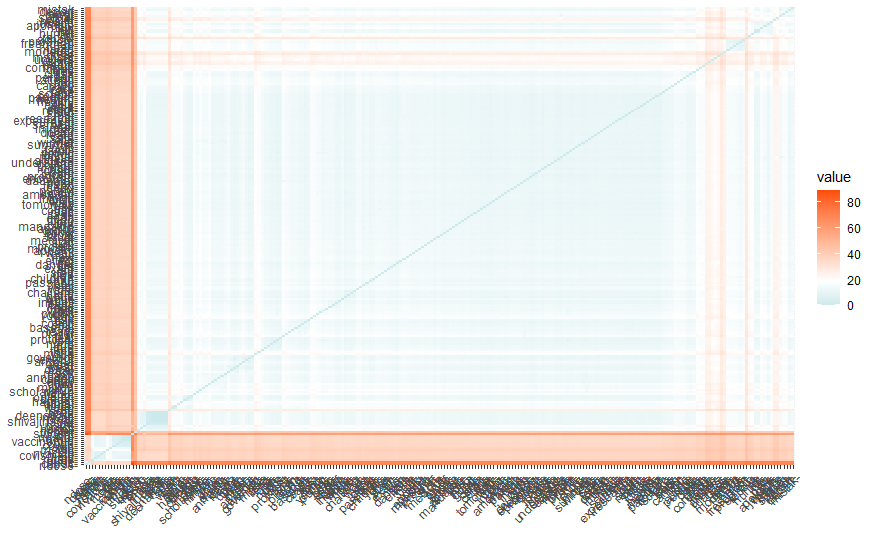
**Restricted Data:**

****

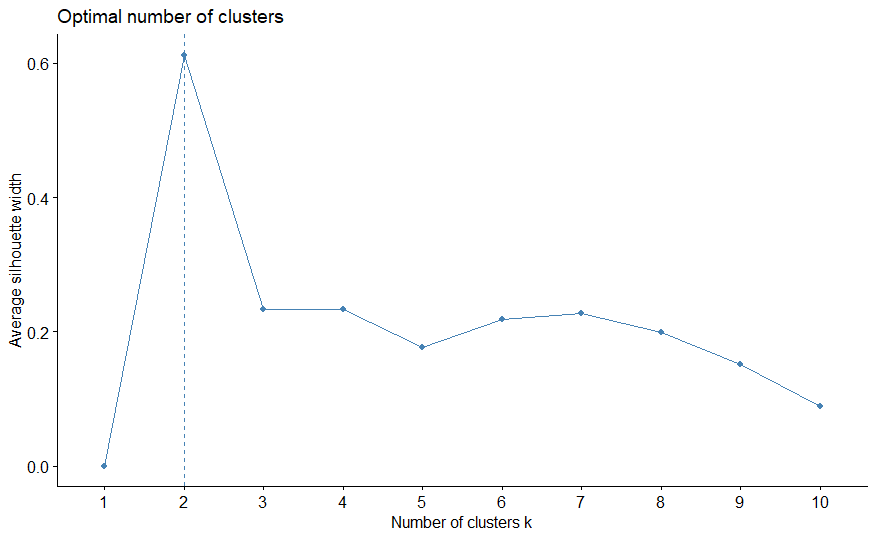
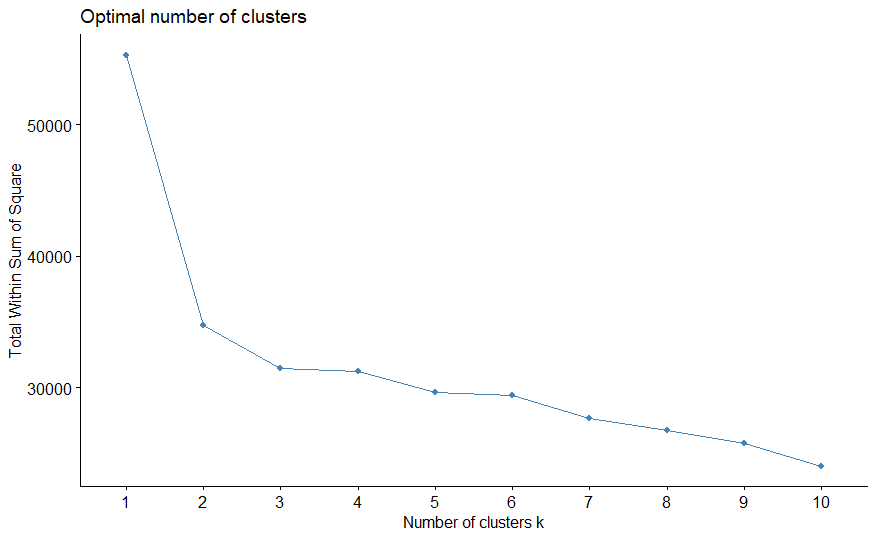
****

**k-Means**

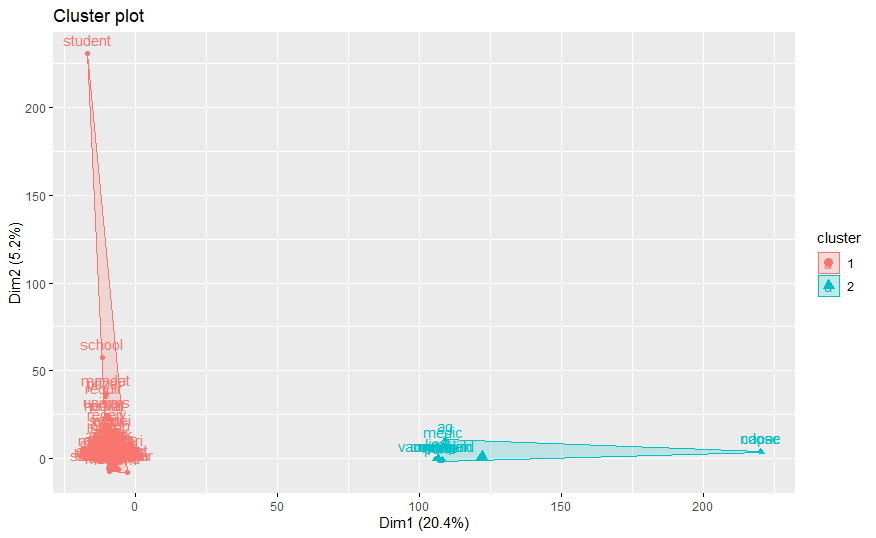
**Whole Data – Heat map**

****

**Whole Data - Words**

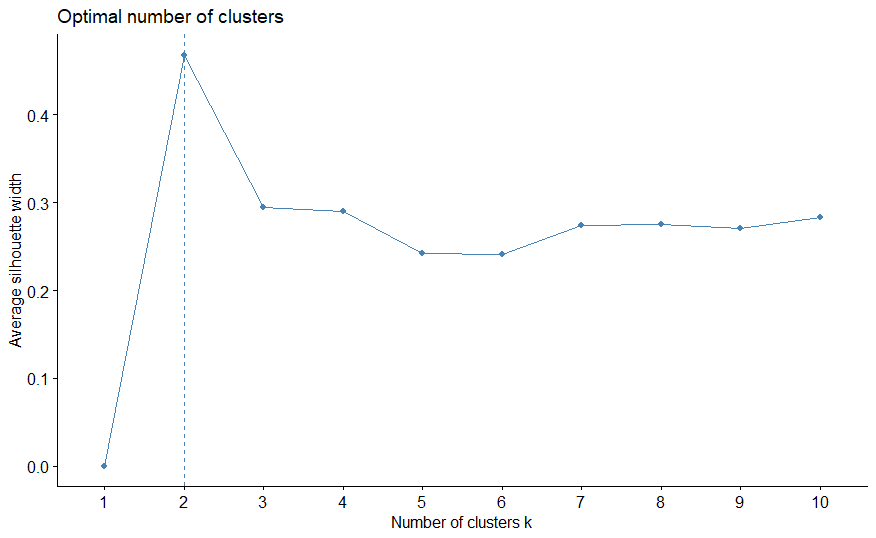
****

Elbow method and silhouette methods respectively for determining number of optimal clusters

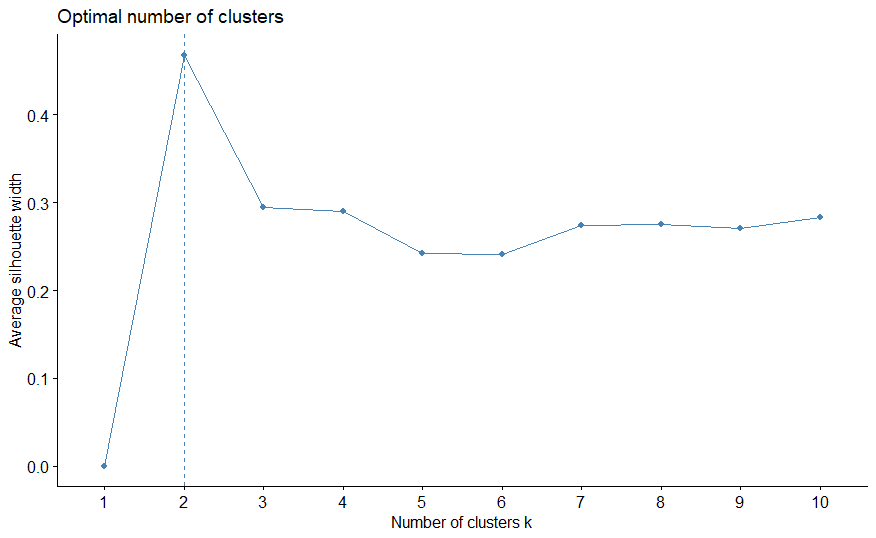


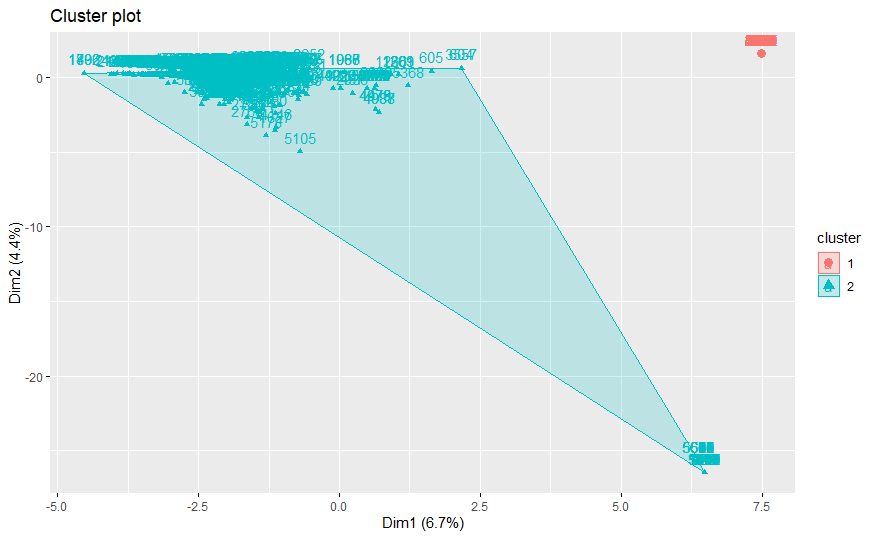
**Whole Data – Tweets:**

Elbow method:

****

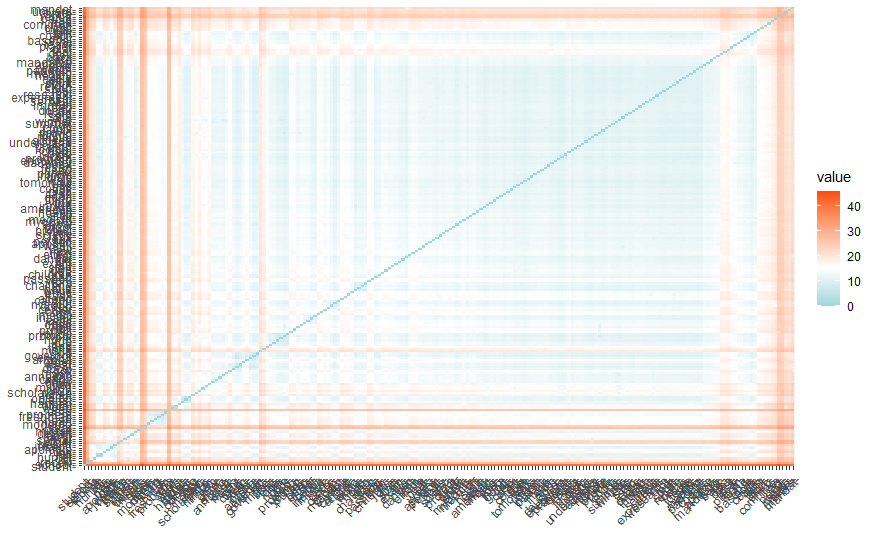
**Silhouette method:**

****

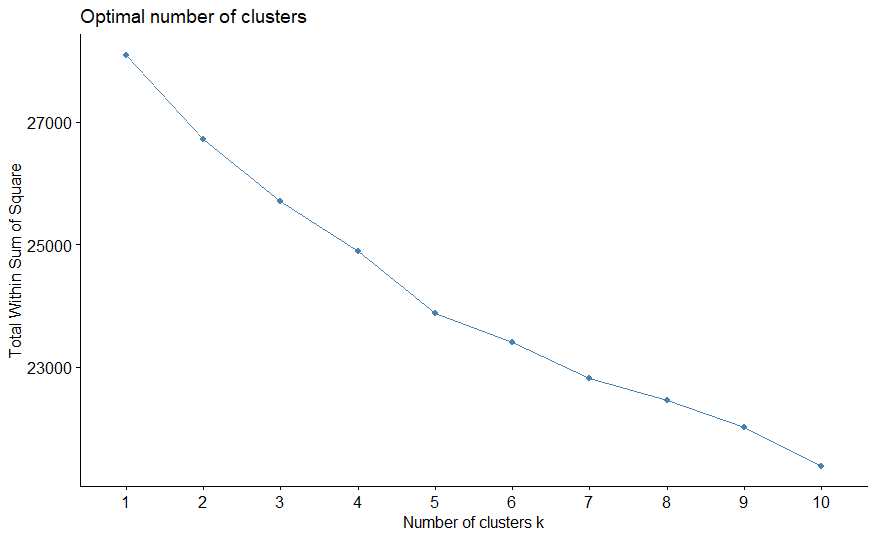
****

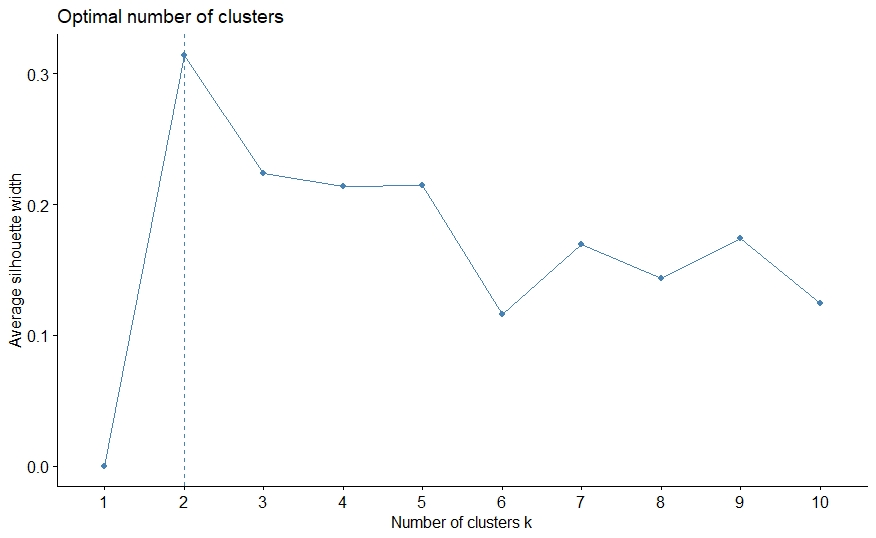
The words in the clusters couldn’t be read properly. So were the tweets. But I think the words in each category represent the broad topic represented by the 2 categories, the tweets have been clustered into.

**Restricted Data – heat map:**

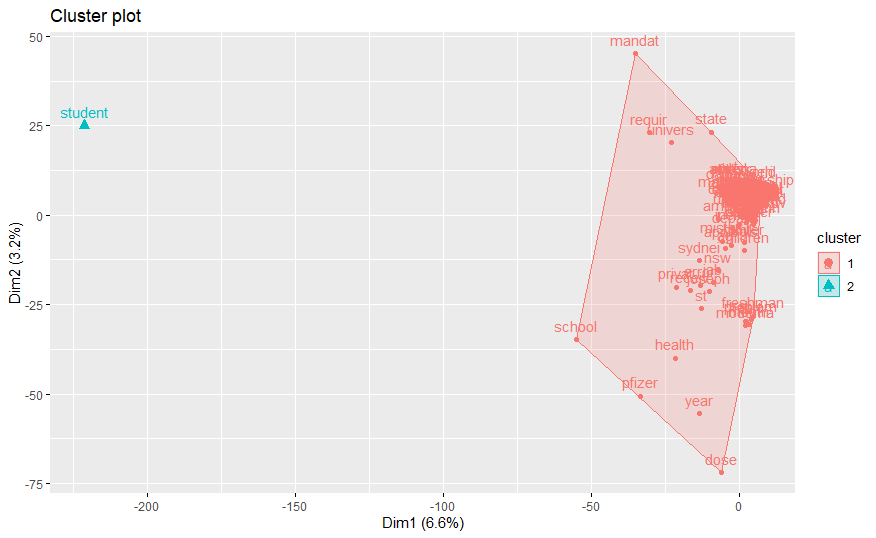
****

**Restricted data - Words:**

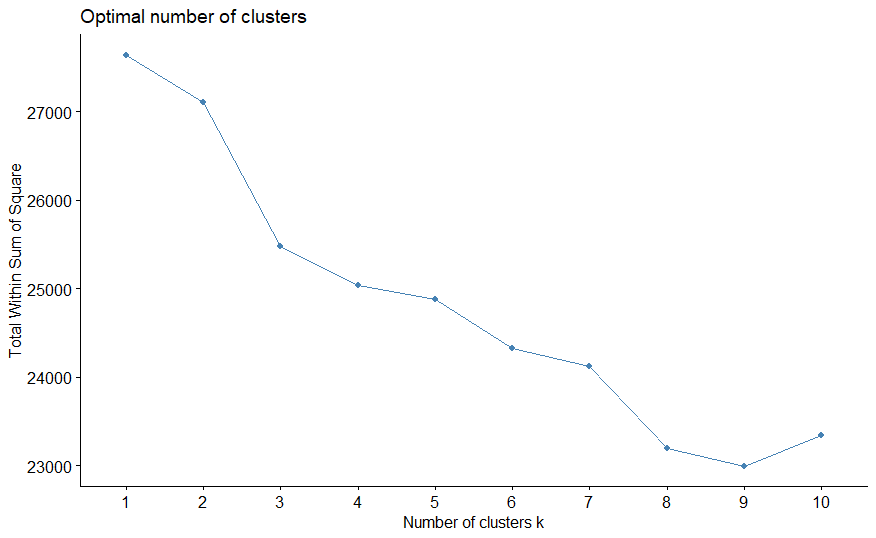
****

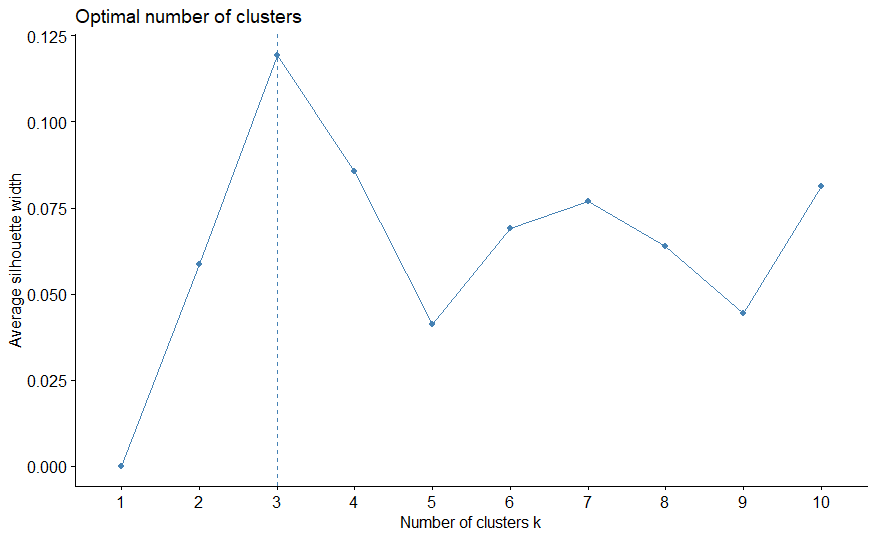
****

Number of clusters not much clear from elbow method. Silhouette suggested 2. So I took 2 clusters.



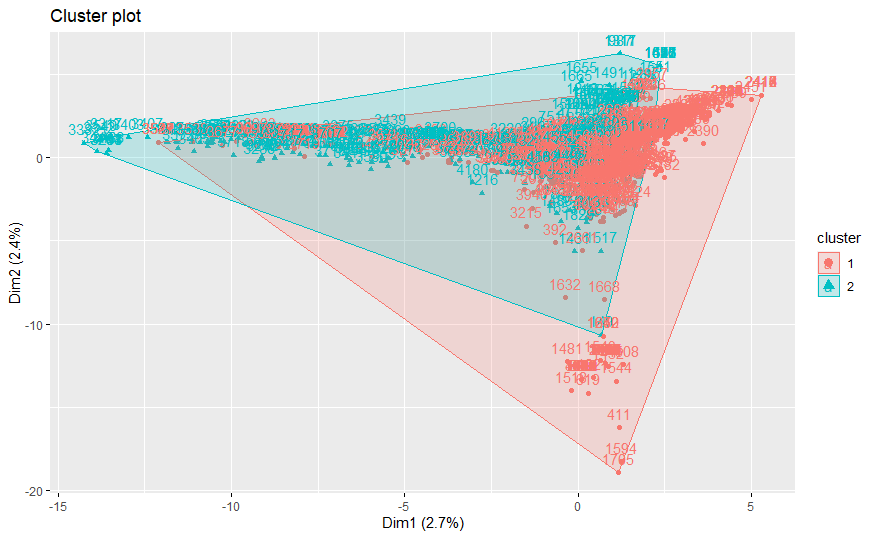
**Restricted Data – Tweets:**

****

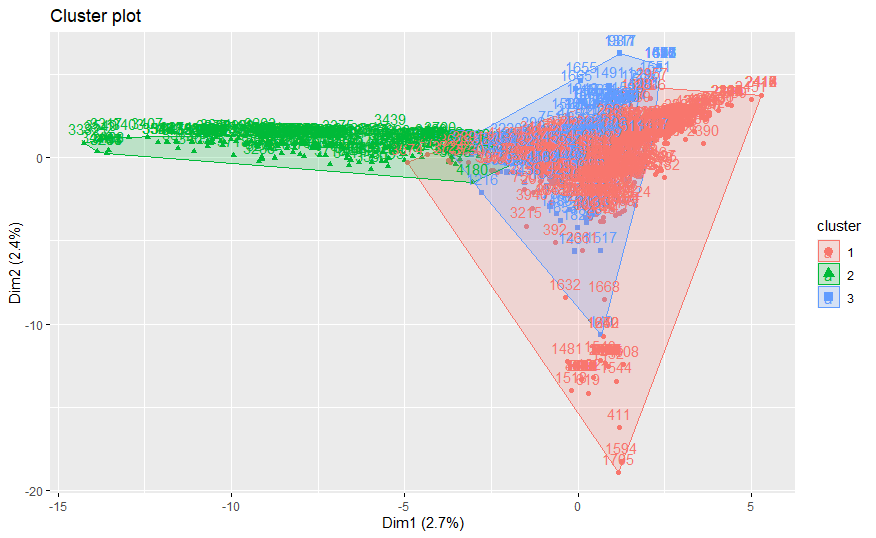
****

**I tried clustering both into 2 and into 3 categoried.**

**2 categories:**

****

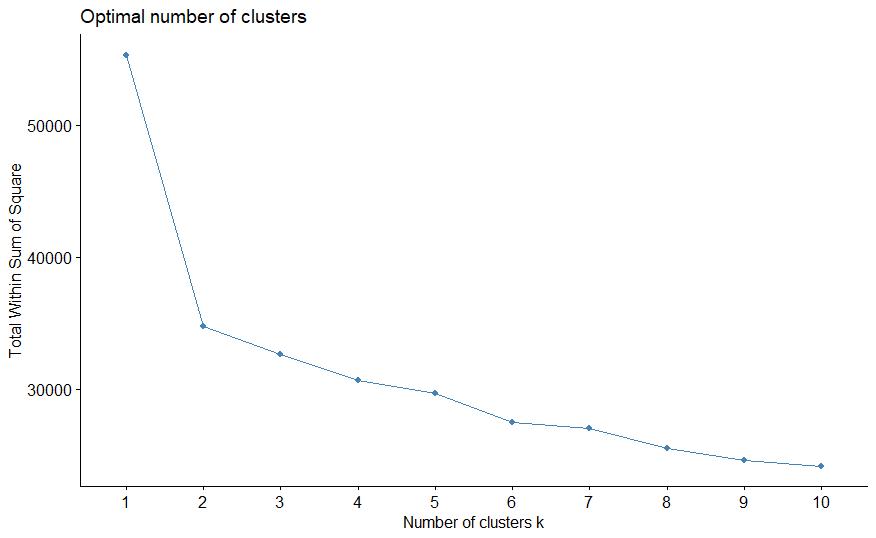
**3 categories:**

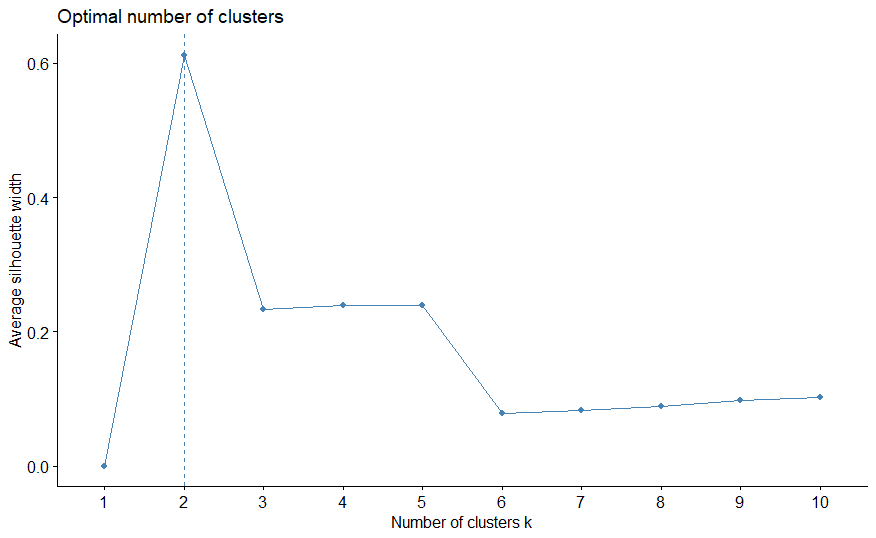
****

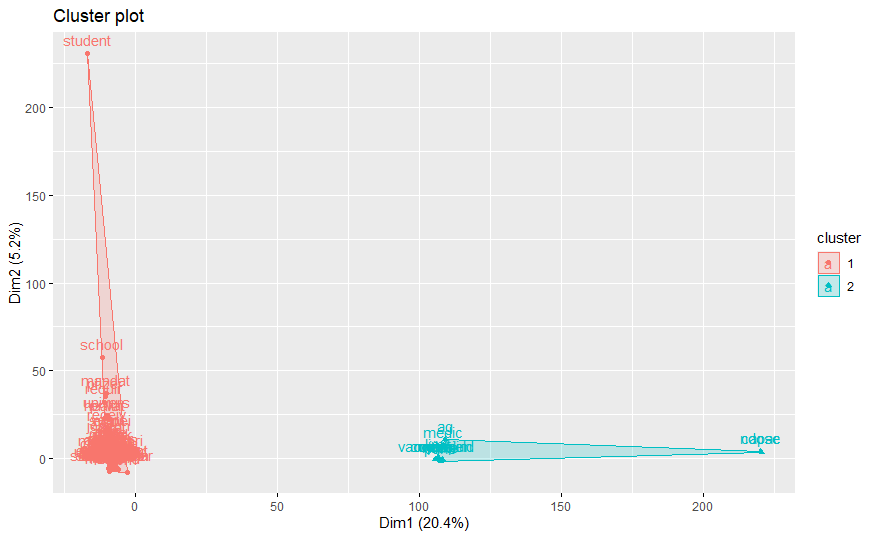
**So clustering into 3 categories was better, it seems from the plots. Also, for restricted tweets, there has been overlap between the categories. I think the pincode tweets had a dominating effect on the number of clusters, ensuring a cluster mostly comprising them. Once they were removed, overlap increases. But with increasing categories from 2 to 3, the overlap somewhat decreased.**

**k-medoids:**

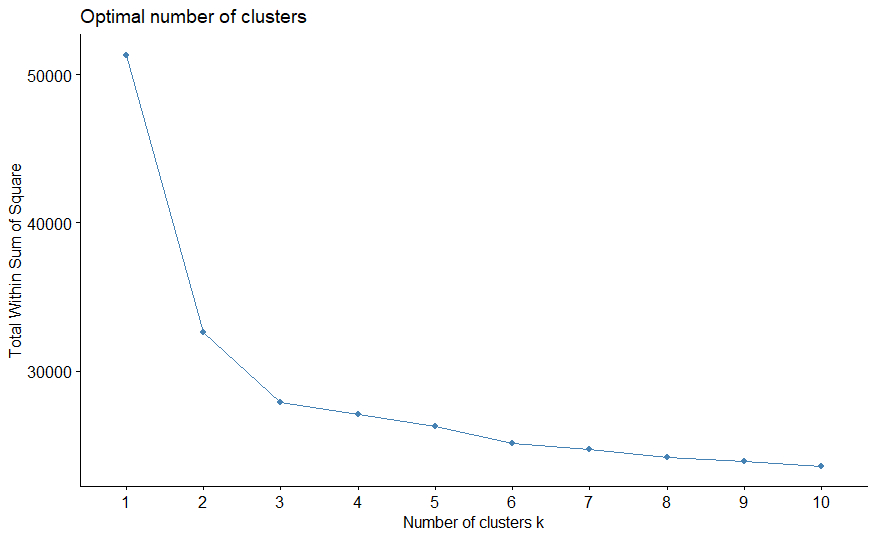
**Whole data – Words**

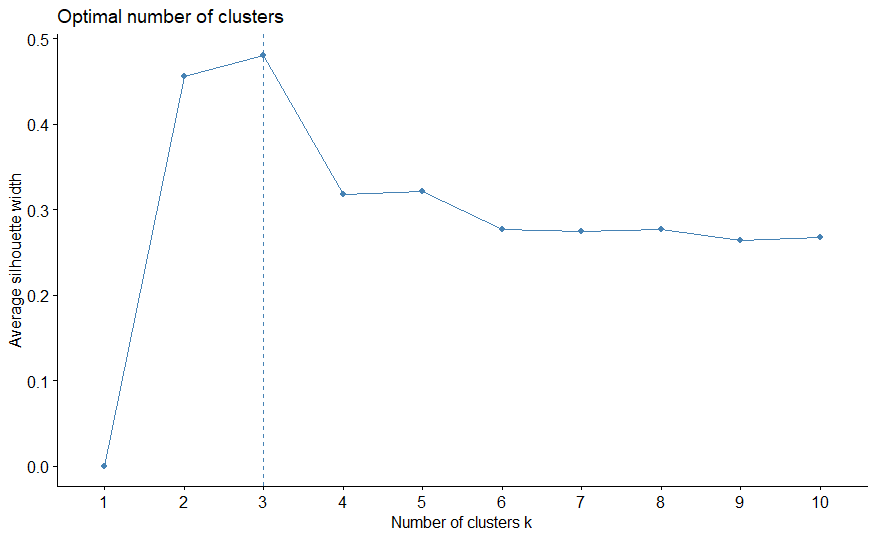
****

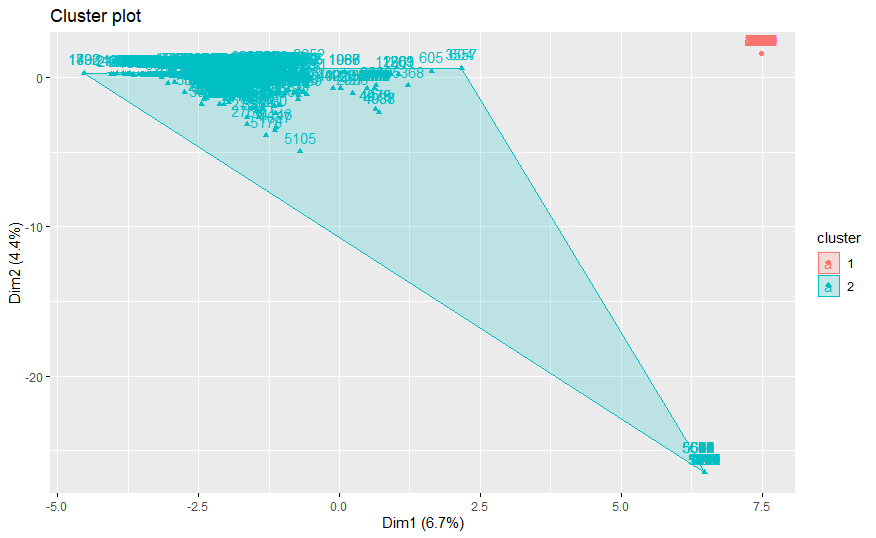
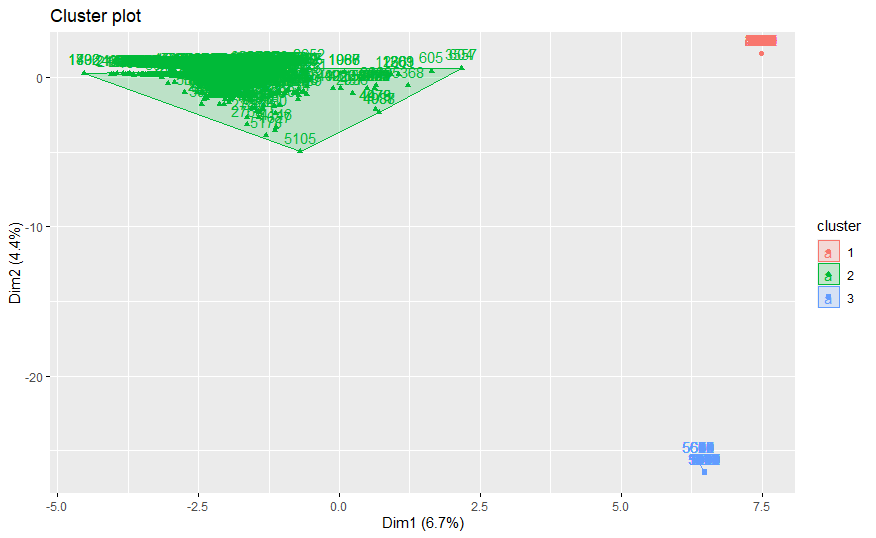
****

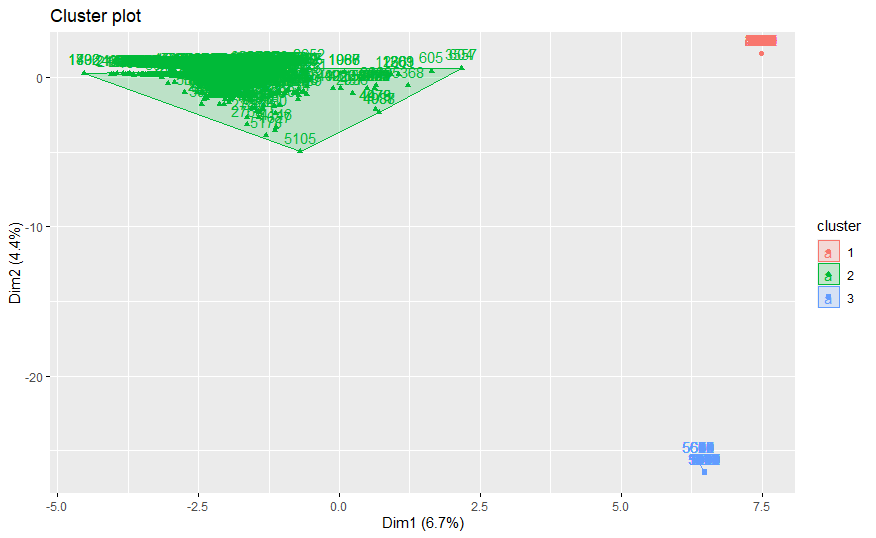
****

**Whole Data – Tweets**

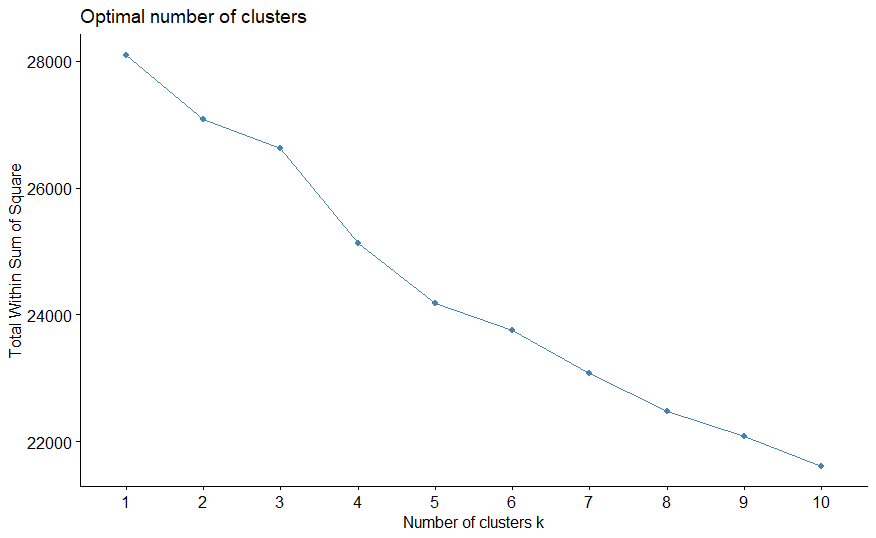
****

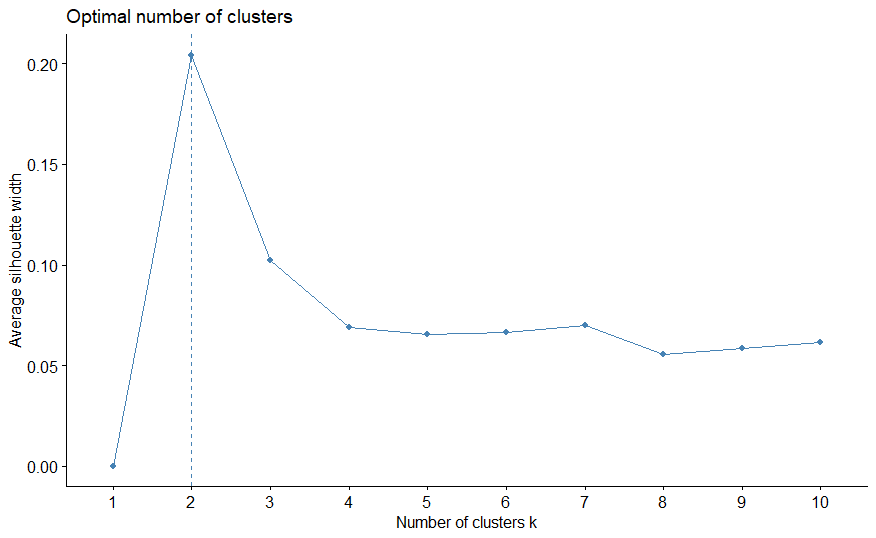
****

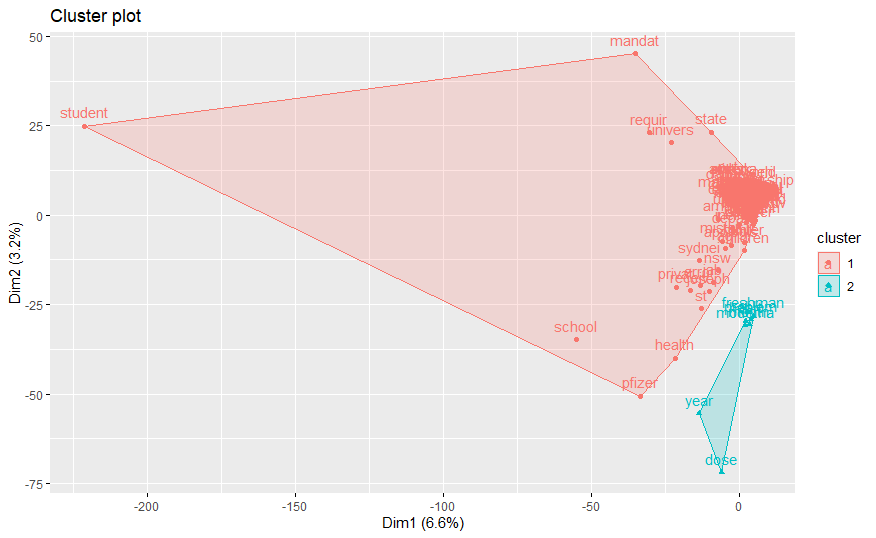
****

****

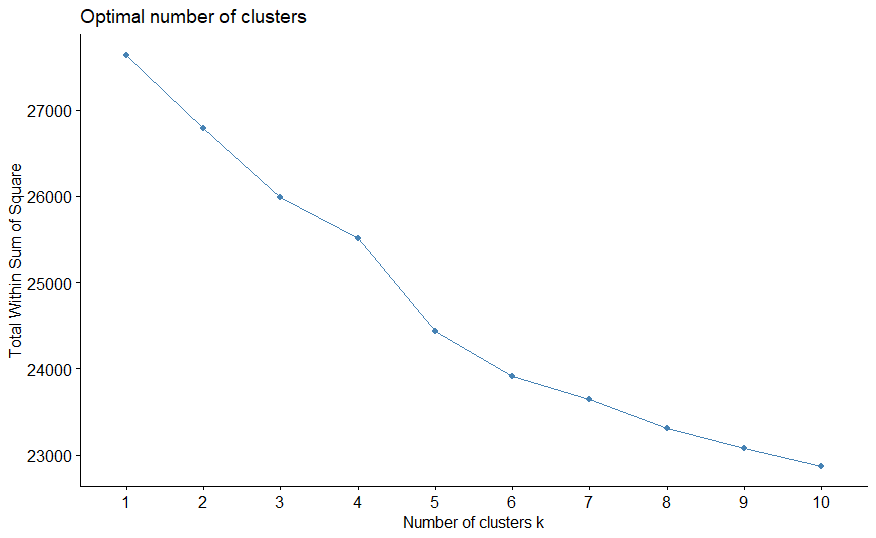
**Restricted Data – Words**

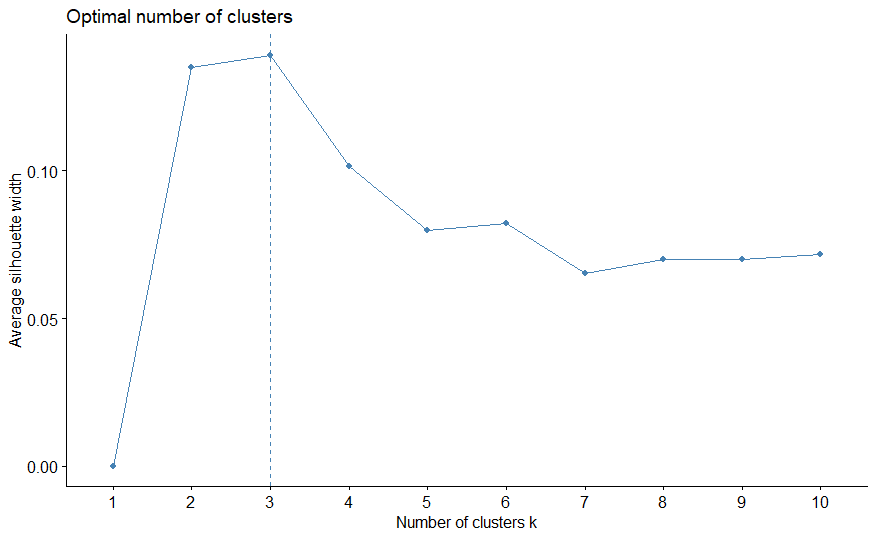
****

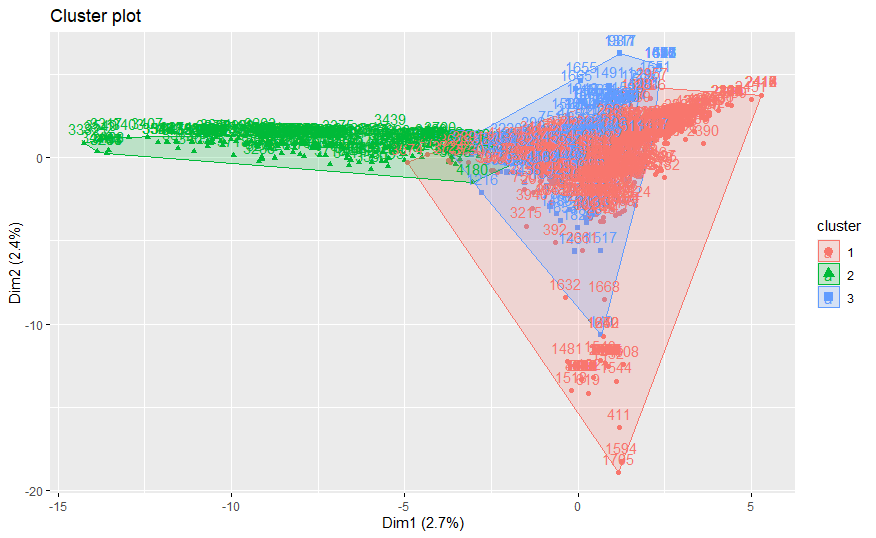
****

****

**Restricted Data- Tweets**

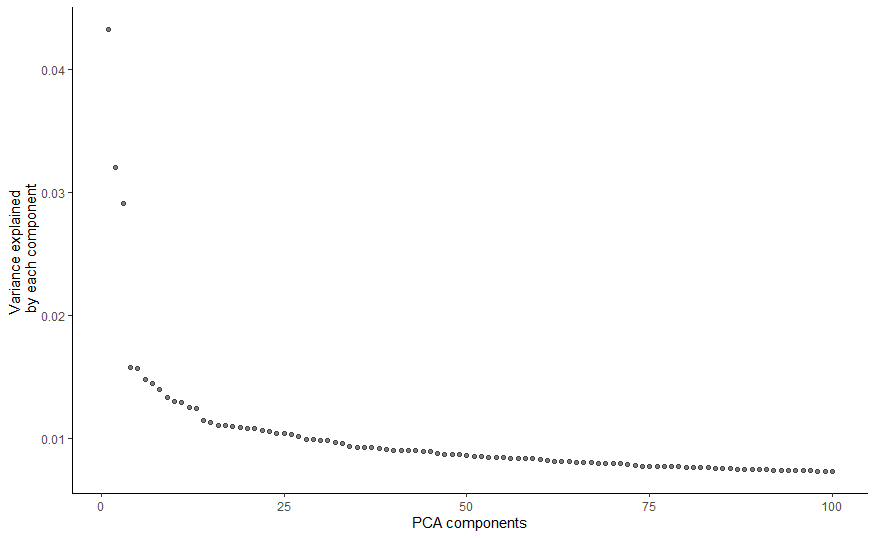
****

****

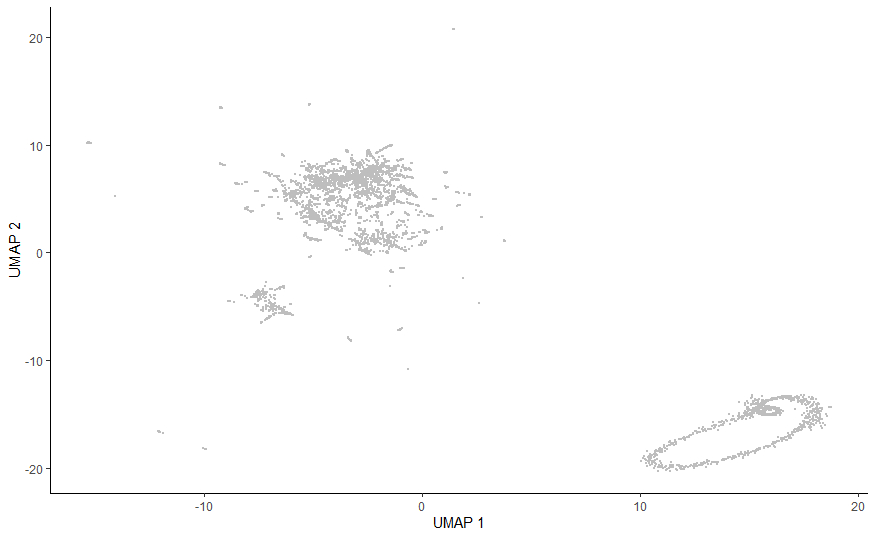
****

So we see, with respect to k-means, the overlap between the 3 categories has decreased in case of k-medoids. So I would say k-medoids performs better than k-means.

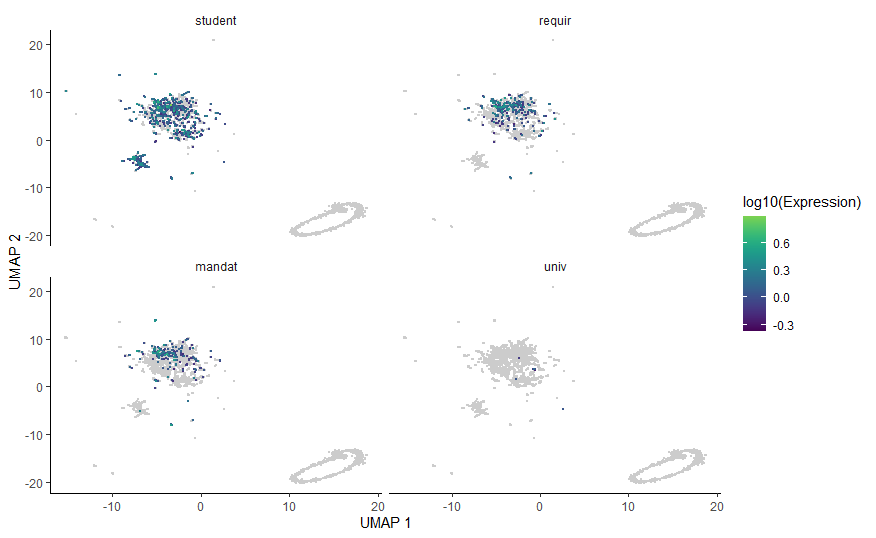
**Monocle3:**

****

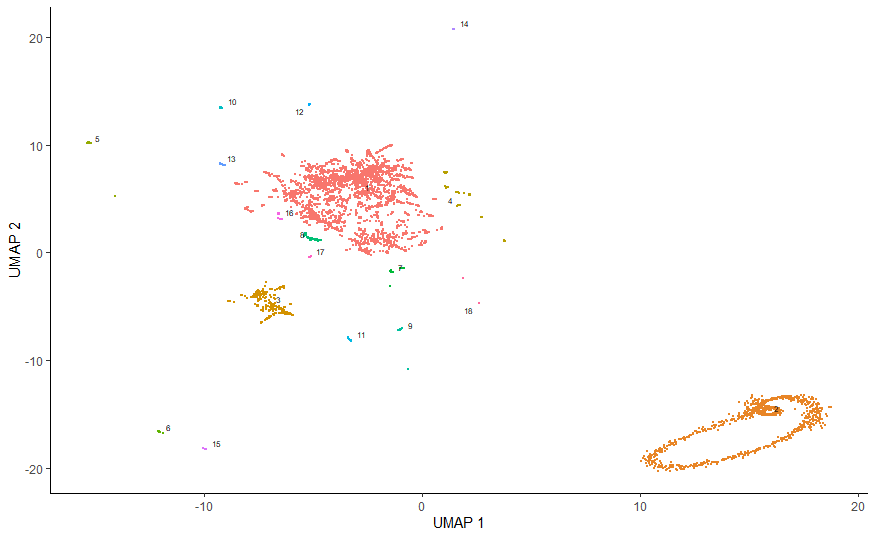
Dimension reduction by PCA. Above plot shows the variance explained by each component. Decreases gradually.

****

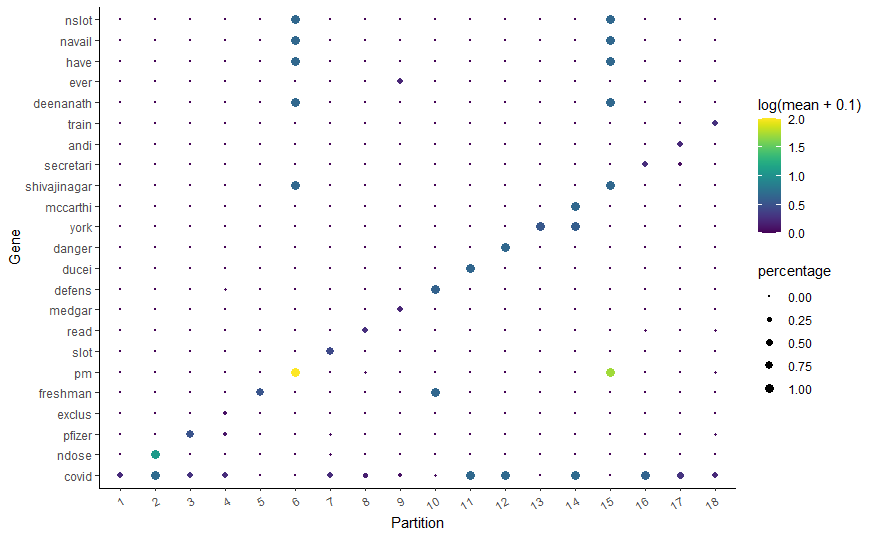
UMAP plot for the tweets after linear and non-linear dimension reduction. Couldn’t colour the tweets by any category like users or anything like that as we don’t have that information. Maybe after calculating vader sentiments, we can colour them by positive and negative tweets.

****

I could remember 4 words: student, require, mandate and univ. So I coloured proportion of these words in the tweets.

****

After clustering I got this. There are many tiny clusters and this has increased the number of clusters to 18 ig. Juejue mentioned DBSCAN to deal with this problem, but couldn’t implement, so I asked her to help. Also for deciding the optimal value for k nearest neighbours in phonograph clustering algorithm of Monocle3, I couldn’t perform silhouette analysis or something like that.

****

Feauture words in each cluster. Monocle3 seems to be best. It classified into so many categories. I think even after removing the minor clusters using DBSCAN, the number of clusters would be more, so chances of any latent topic being left out will also be less. I am not sure if I am right. Please share what you think of this analysis. And if you guys feel like performing monocle3 for further clustering, it would be great if you guys can come up with solutions for the DBSCAN and silhouette analysis problems. I have also mailed Juejue regarding this.