HW7_Responses

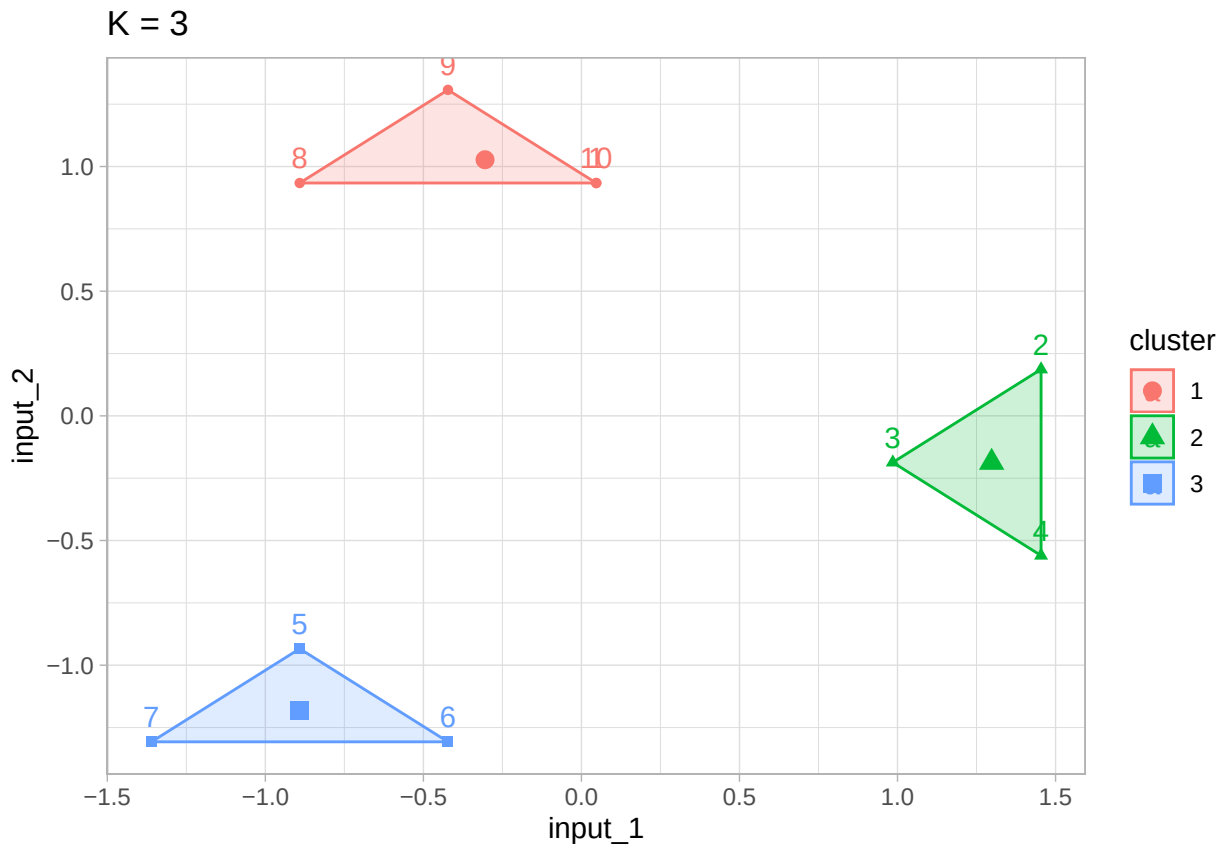
Yawei LI

3/15/2020

K-Means Clustering by Hand

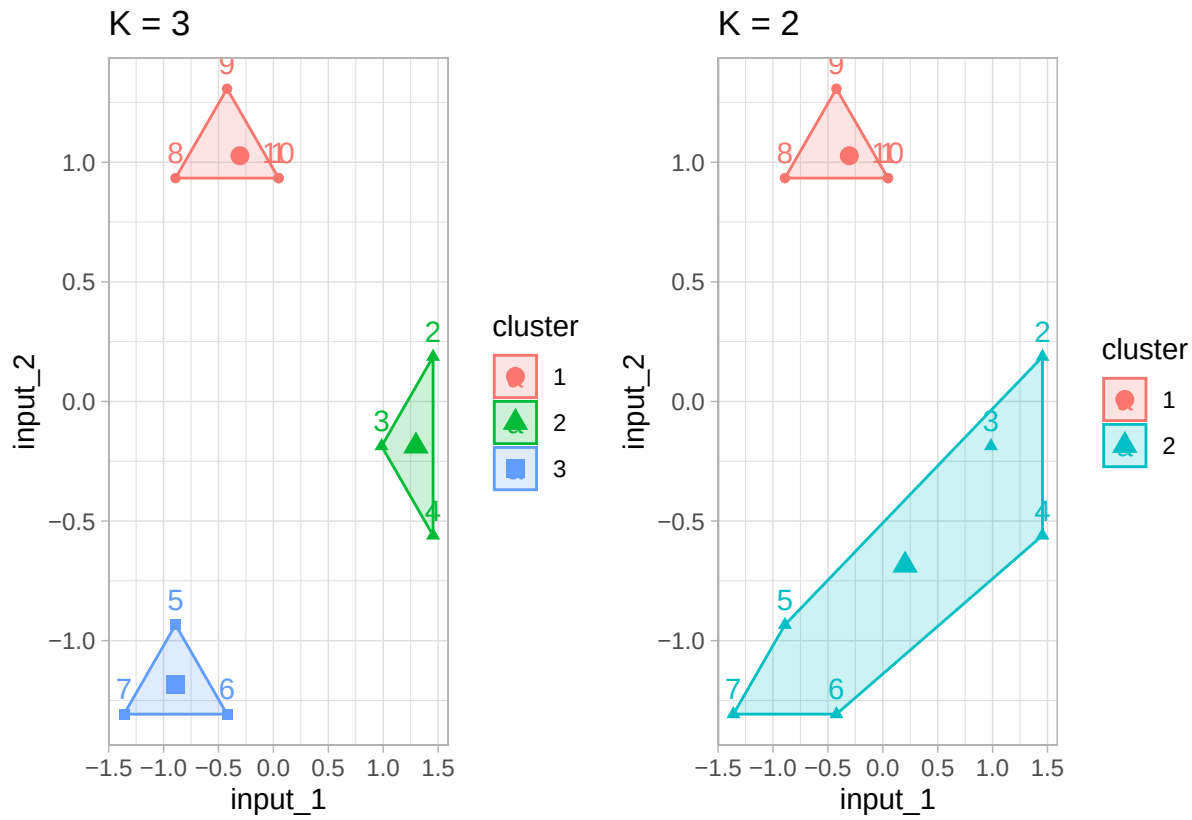
K = 3

```
set.seed(1)
input_1 = c(5,8,7,8,3,4,2,3,4,5)
input_2 = c(8,6,5,4,3,2,2,8,9,8)
df = data.frame(input_1,input_2)
k3 = kmeans(df,centers = 3, nstart = 1)
p1 = fviz_cluster(k3, data = df) + theme_light() + labs(title = "K = 3")
p1
```



K=2

```
set.seed(2)
k2 = kmeans(df,centers = 2, nstart = 1)
p2 = fviz_cluster(k2, data = df) + theme_light() + labs(title = "K = 2")
p1+p2
```



Discussion Obviously, K=3 did a better job. In K=2 plot, the withiness of Cluster 2 is way larger than Cluter2 and Cluster3 combined. K=3 provides a much more compact plot.

Dimension Reduction

Data Prep

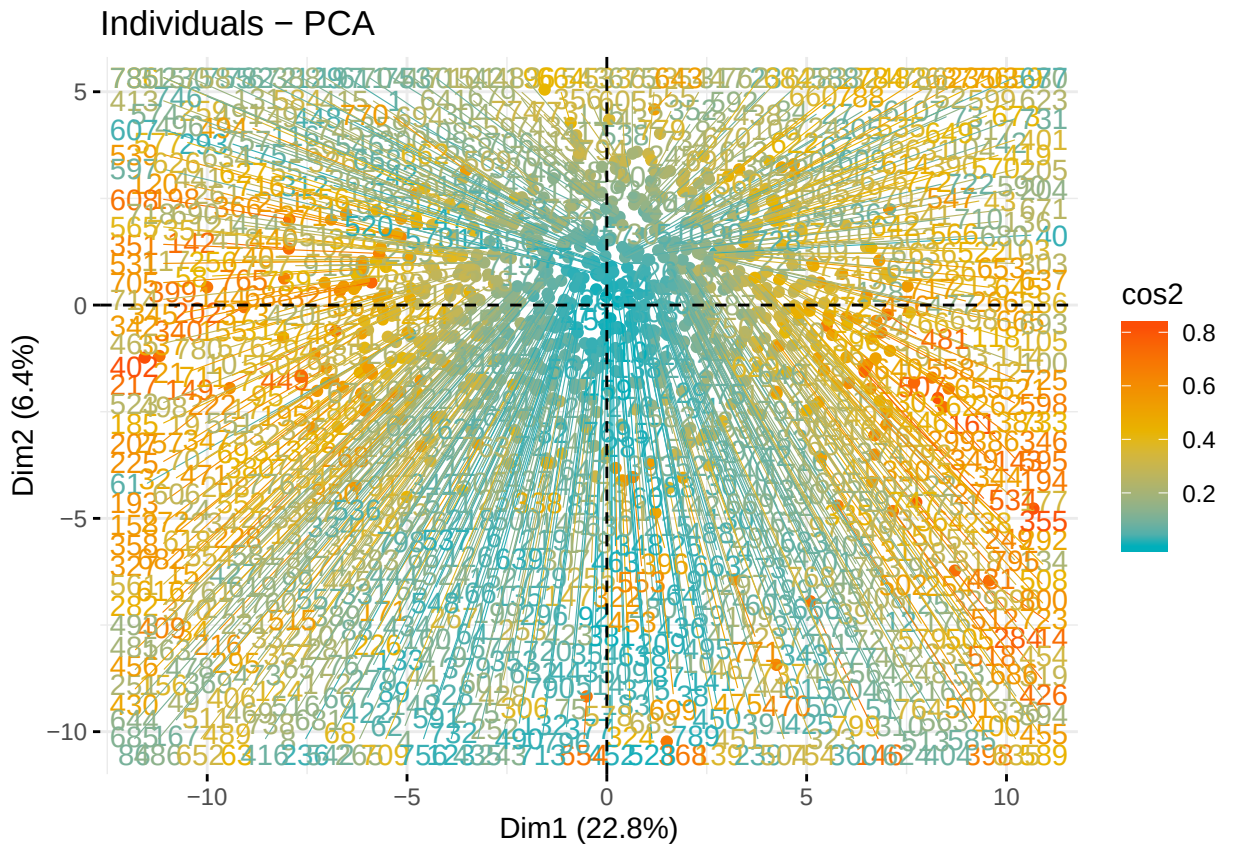
```
wiki = read.csv('data7/wiki.csv')
```

PCA

```
wiki_pca <- prcomp(wiki, center = TRUE, scale = TRUE)
```

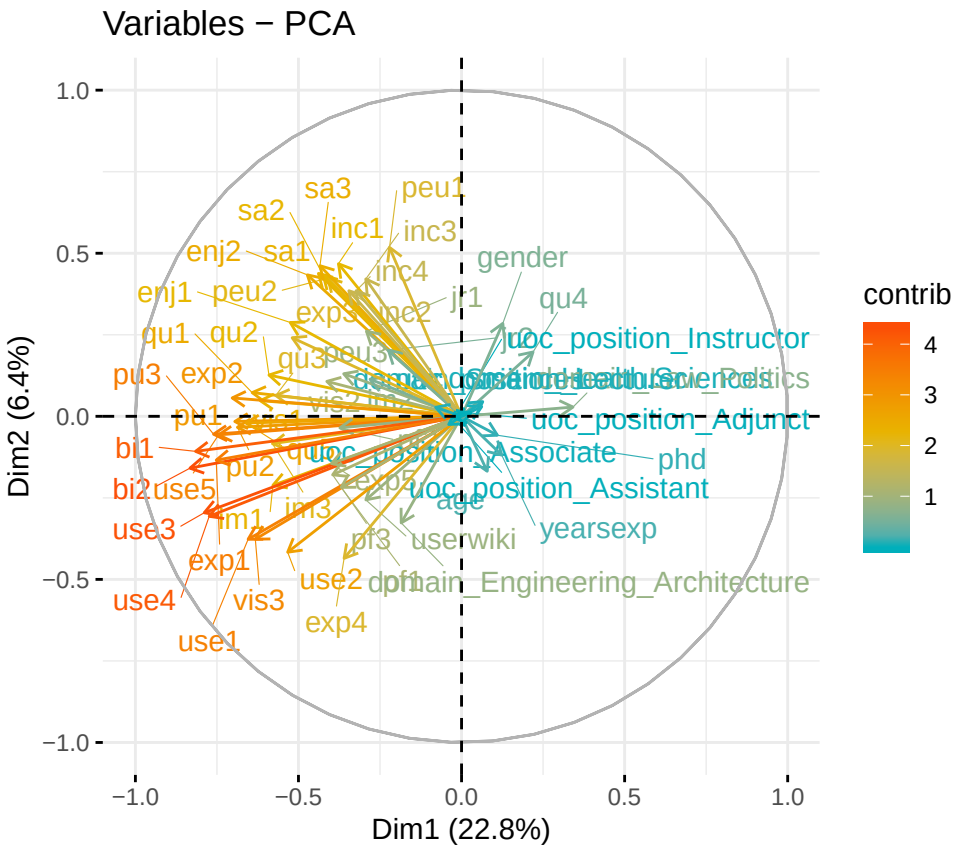
Individuals

```
wiki.ind <- get_pca_ind(wiki_pca)
fviz_pca_ind(wiki_pca,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



Variables

```
wiki.var <- get_pca_var(wiki_pca)
fviz_pca_var(wiki_pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



Discussion

As can be seen from the variable plot, bi, pu, use and exp (except exp4) are strongly correlated to the second principal component. The correlations on the first component are not as strong, yet sa, inc and peu are well correlated for PC1.

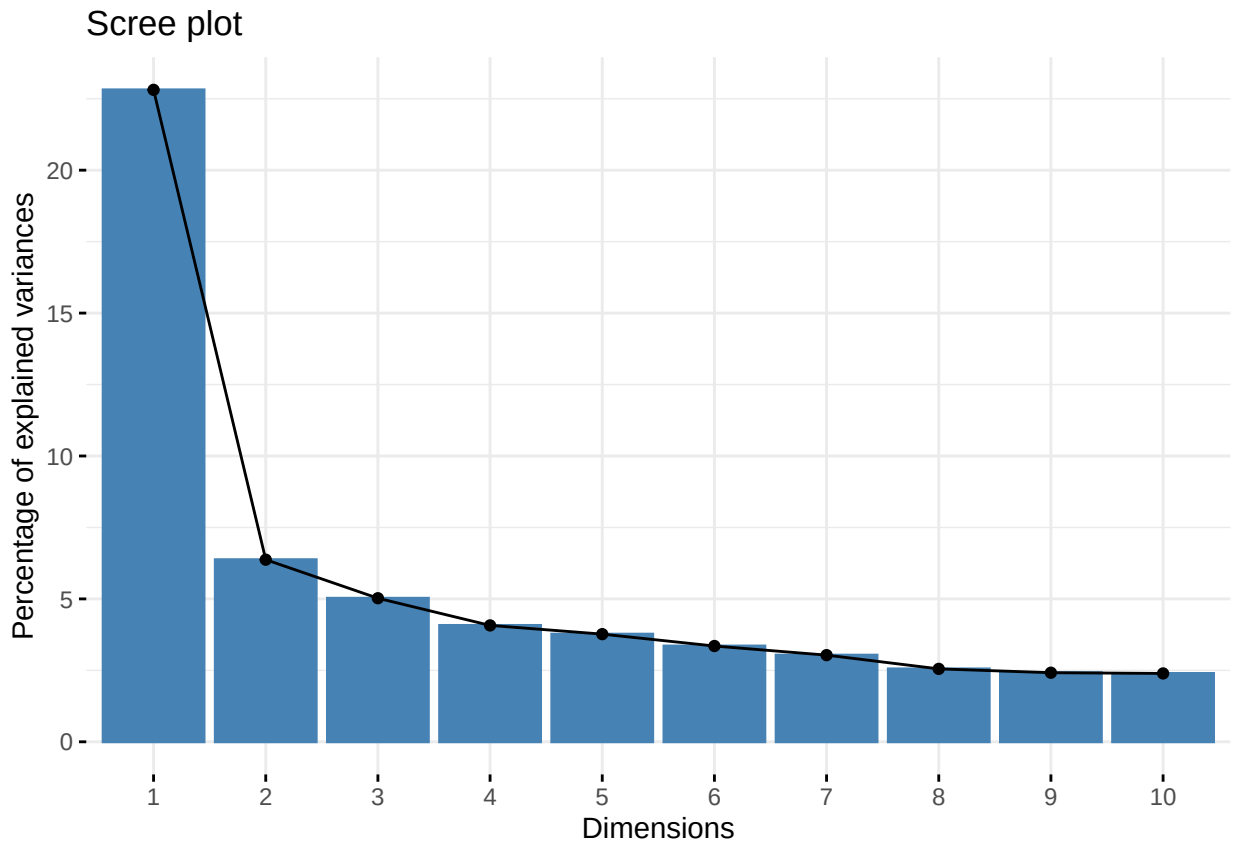
PVEs

The first two components explained only 29.1% of the variance.

```
pr_var = data.frame(varExp = wiki_pca$sdev^2)
pr_var = pr_var %>%
  mutate(pve = varExp / sum(varExp))
pr_var[1:10,]
```

```
##       varExp        pve
## 1  13.002058 0.22810628
## 2   3.632310 0.06372475
## 3   2.863513 0.05023707
## 4   2.321516 0.04072835
## 5   2.147602 0.03767724
## 6   1.910693 0.03352093
## 7   1.728889 0.03033138
## 8   1.454741 0.02552178
## 9   1.377933 0.02417427
## 10  1.363733 0.02392515
```

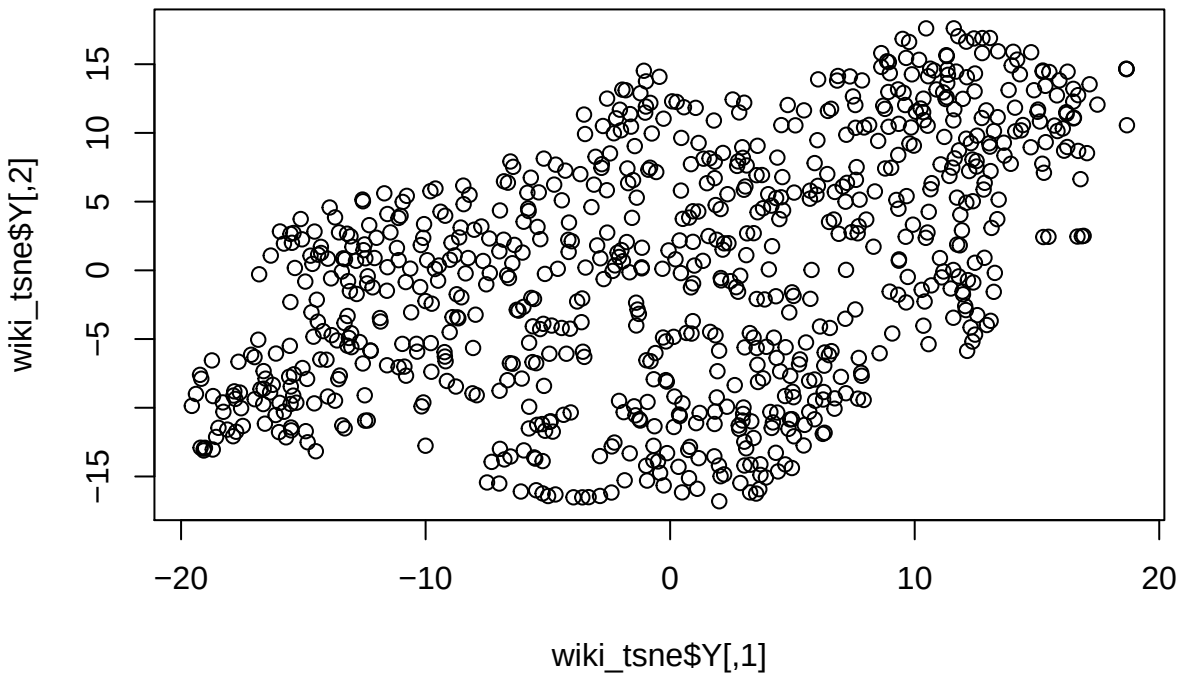
```
fviz_eig(wiki_pca)
```



tSNE

Below is the plot of t-SNE performed. From my best knowledge, not much insight can be gained from the projected observation data alone, nor do I know how to colorise it properly for clarification. It appears that our data is relatively evenly distributed across the two dimensions, with observations where Dim1<0 and Dim2 > 0 rather few (upperleft in plot).

```
wiki_tsne <- Rtsne(wiki, dims = 2)
p_tse = plot(wiki_tsne$Y)
```

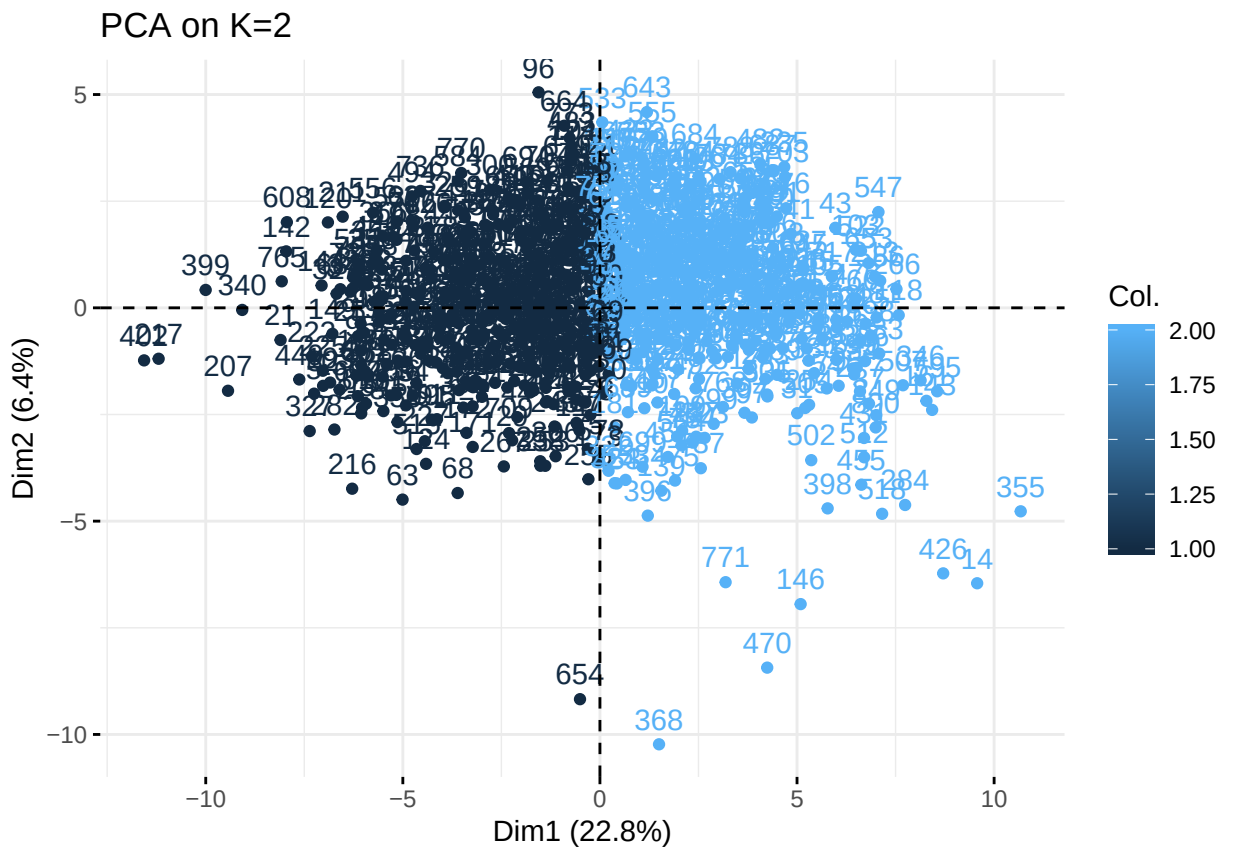


Clustering

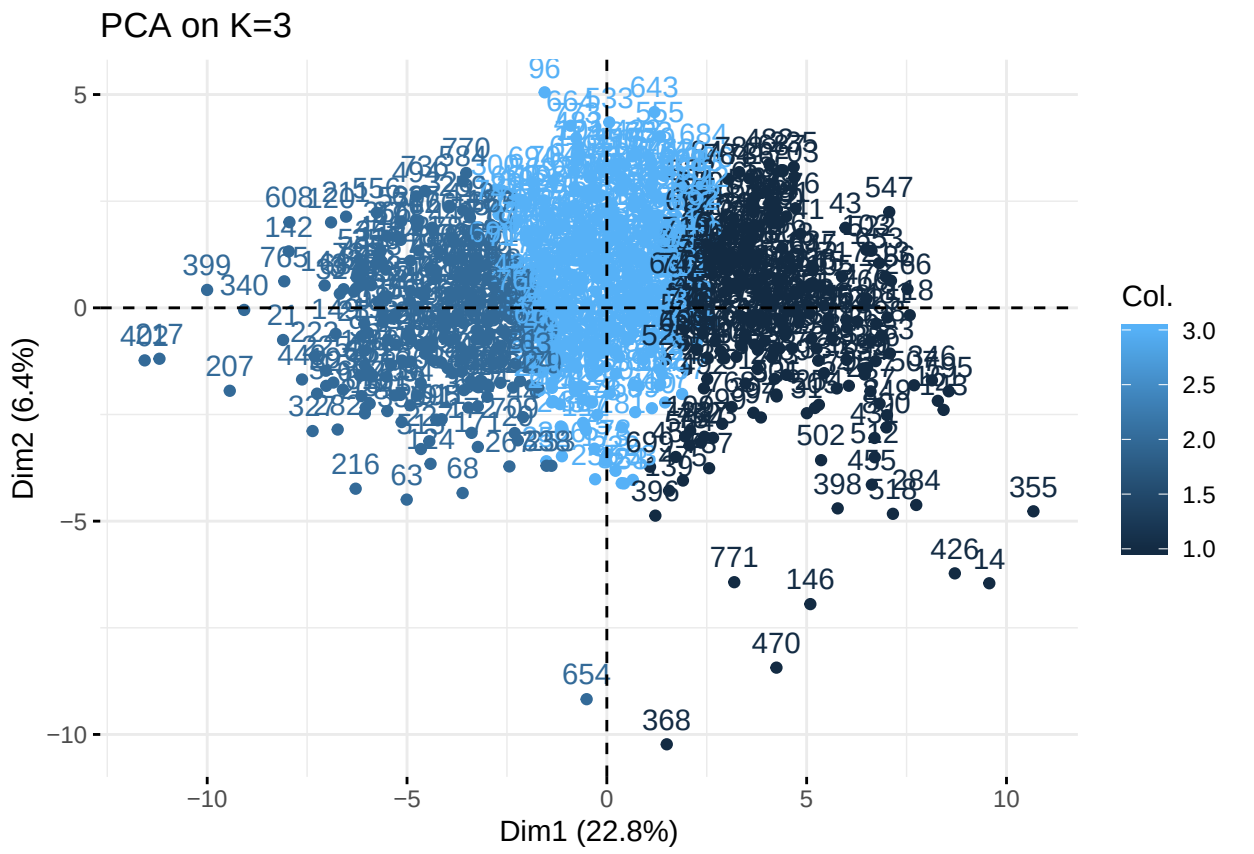
K-means

As can be seen from the plot, the principal component analysis did well in seperating the clusters, at K = 2 the two clusters falls apart between the first primal component. However, it can be seen from the plots that when K>2 it becomes harder to explain. to interpret intuitively.

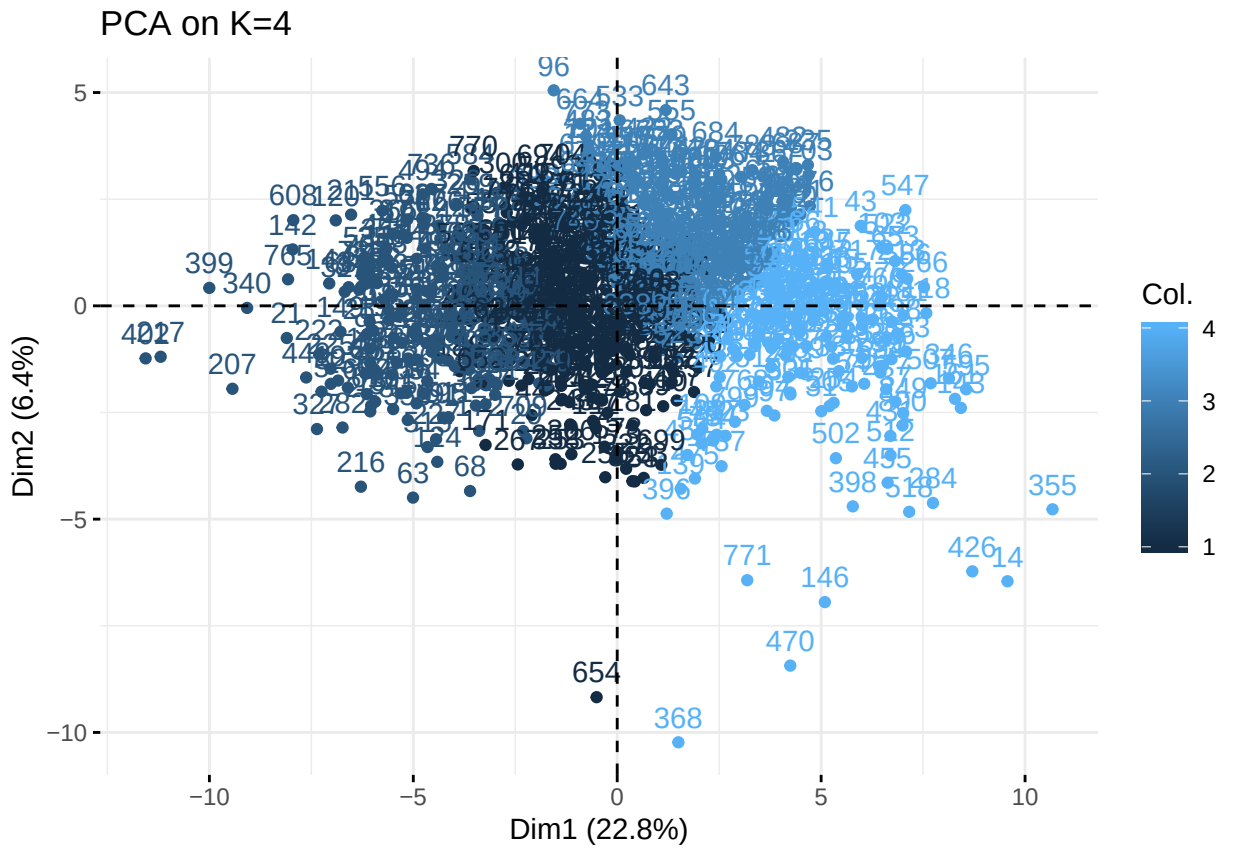
```
set.seed(3)
wiki_s = scale(wiki)
k2 = kmeans(wiki_s,2,nstart = 20)
k3 = kmeans(wiki_s,3,nstart = 20,)
k4 = kmeans(wiki_s,4,nstart = 20,)

p1 = fviz_pca_ind(wiki_pca,col.ind = k2$cluster) + labs(title = 'PCA on K=2')
p2 = fviz_pca_ind(wiki_pca,col.ind = k3$cluster) + labs(title = 'PCA on K=3')
p3 = fviz_pca_ind(wiki_pca,col.ind = k4$cluster) + labs(title = 'PCA on K=4')
p1
```


PCA on K=2
Dim2 (6.4%)
Dim1 (22.8%)
Col.
2.00
1.75
1.50
1.25
1.00
5
0
−5
−10
−10
−5
0
5
10

```
p2
```



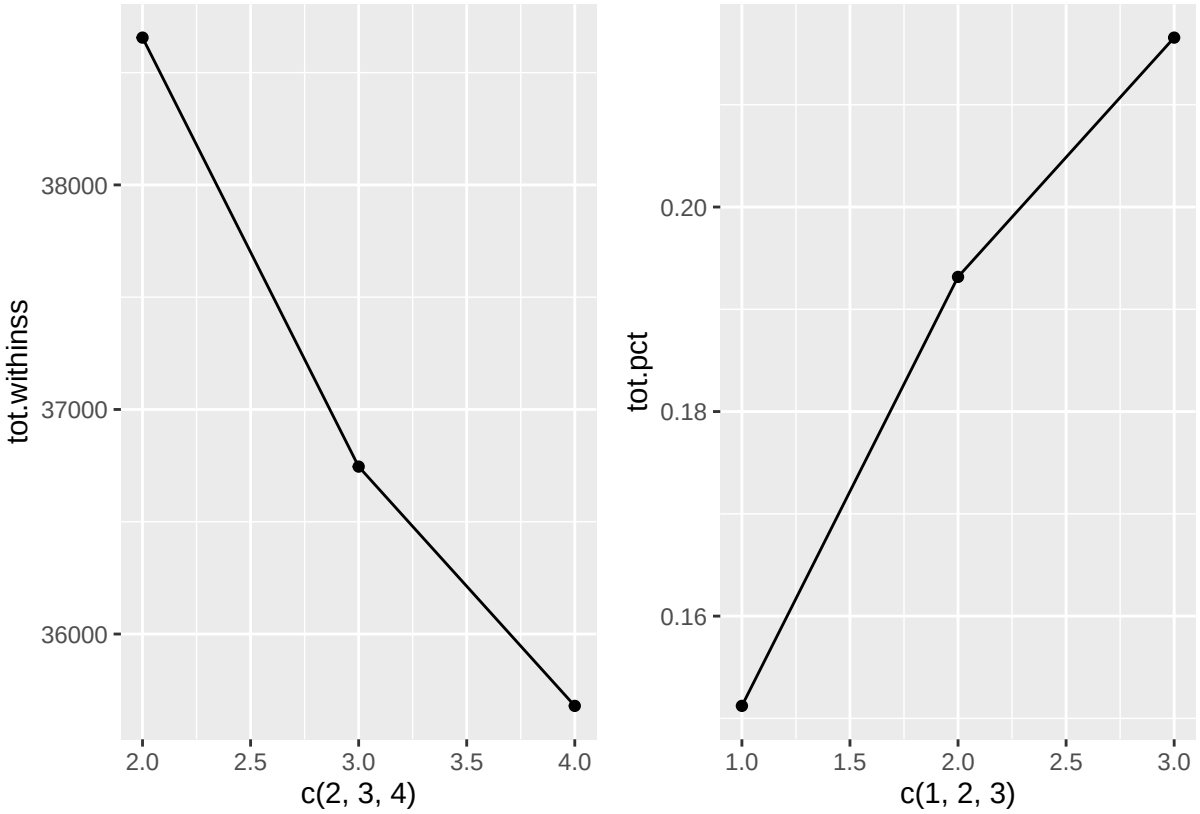

```
p3
```



Identify Best K

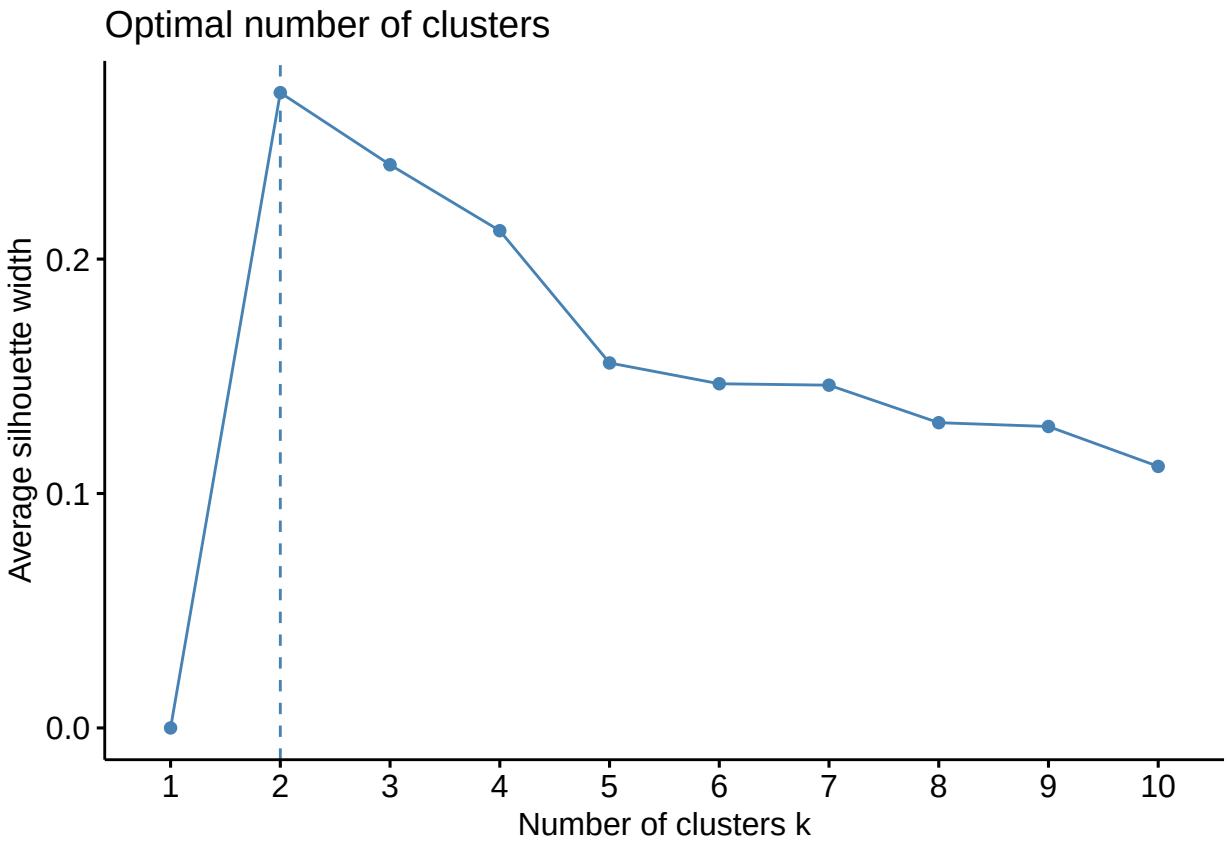
From the left plot of total within sum of squares, it can be seen that K=3 isn't a very good elbow point to choose. In the right plot of betweenss/total ss, K=2 performs the best and it is also the simplest.

```
tot.withinss = c(k2$tot.withinss,k3$tot.withinss,k4$tot.withinss)
tot.pct = c(k2$betweenss/k2$totss,k3$betweenss/k3$totss,k4$betweenss/k4$totss)
evaluation = data.frame(tot.withinss, tot.pct)
p4 = ggplot(data = evaluation, aes(y = tot.withinss, x = c(2,3,4)))
p4 = p4 + geom_line() + geom_point()
p5 = ggplot(data = evaluation, aes(y = tot.pct, x = c(1,2,3)))
p5 = p5 + geom_line() + geom_point()
p4 + p5
```

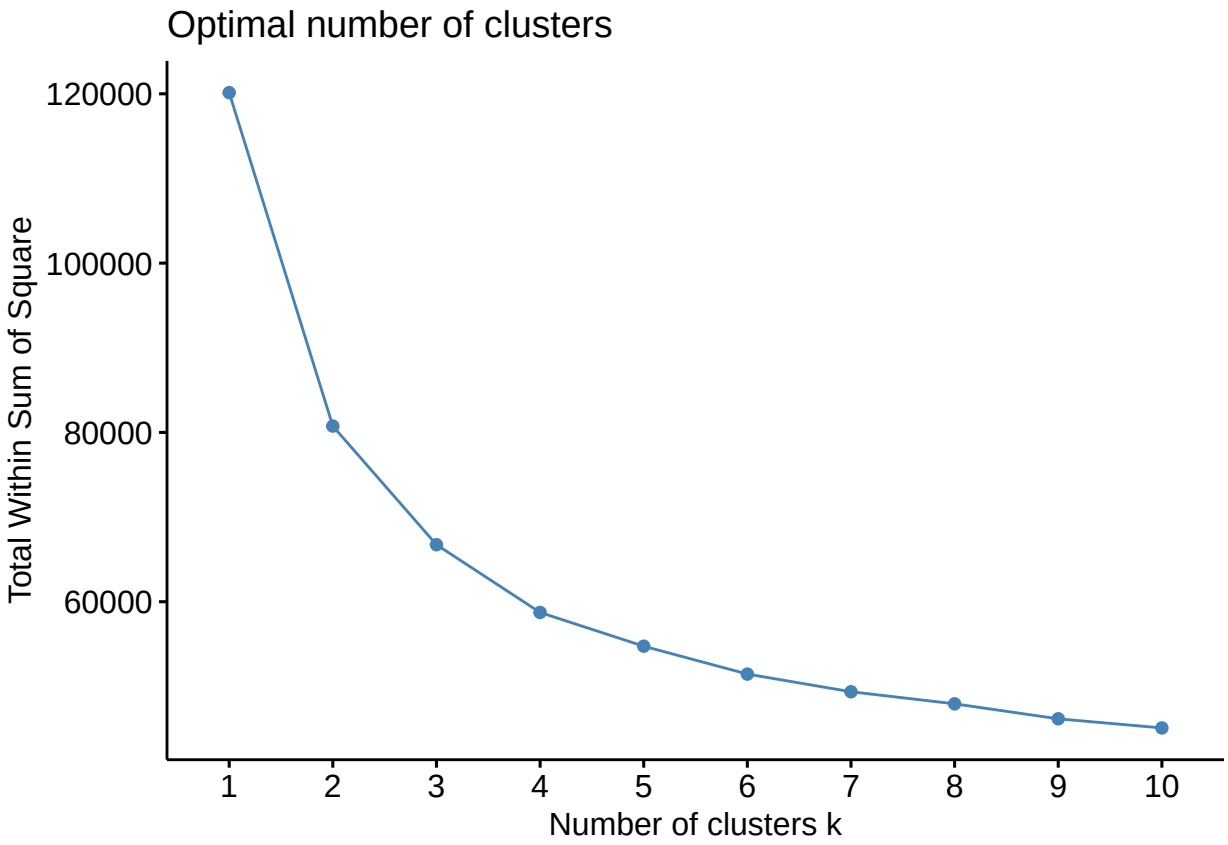


Here I found factoextra has built-in methods to one-line these plots. They also suggest that K=2 is the optimal choice.

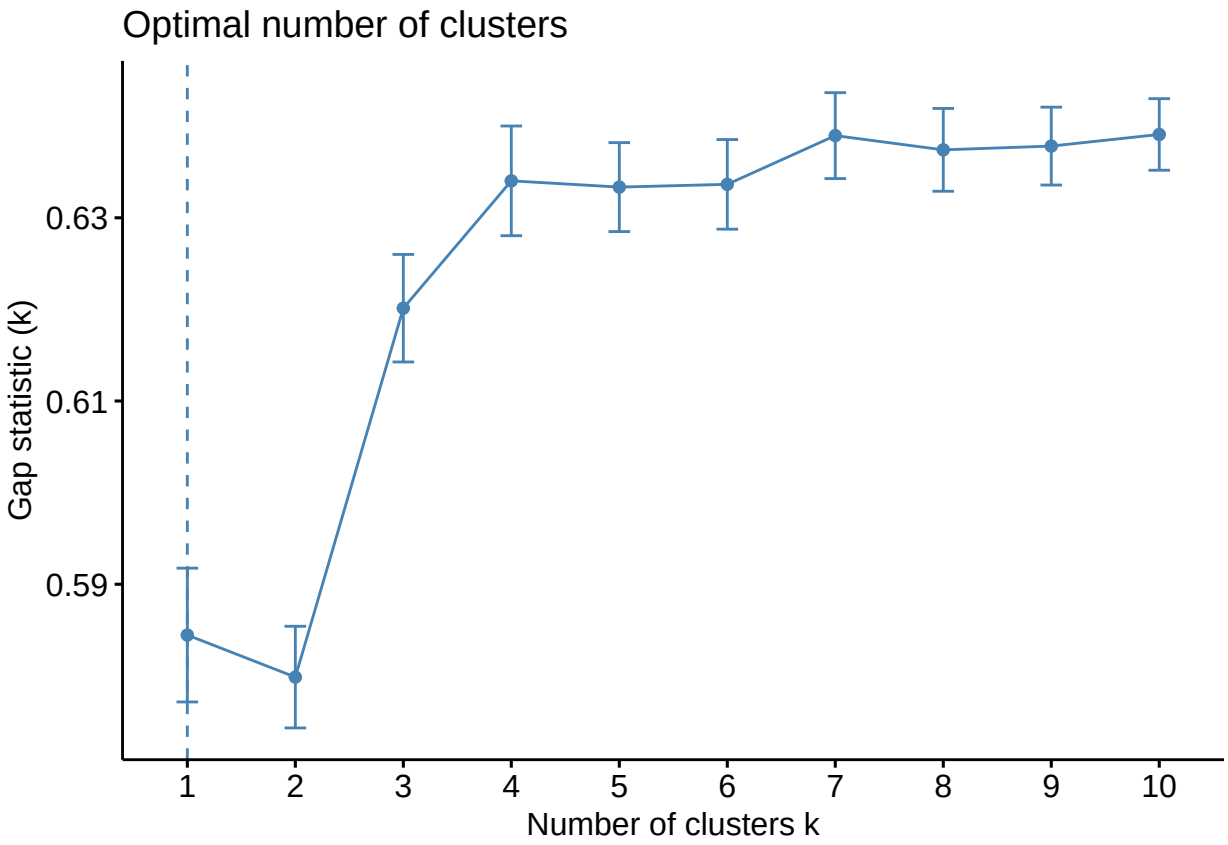
```
fviz_nbclust(wiki,kmeans, method = 'silhouette')
```




```
fviz_nbclust(wiki,kmeans, method = 'wss')
```



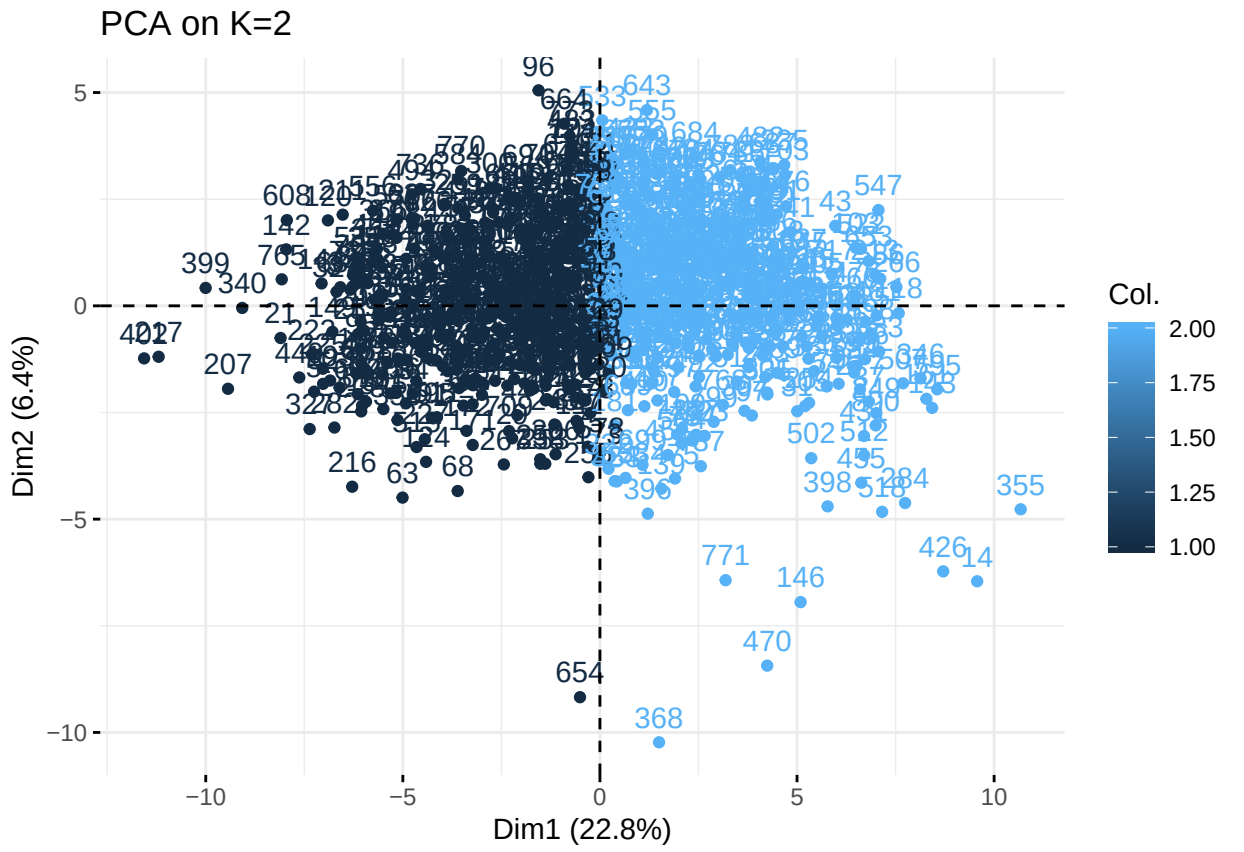

```
fviz_nbclust(wiki,kmeans, method = 'gap_stat')
```



Plot the best clustering

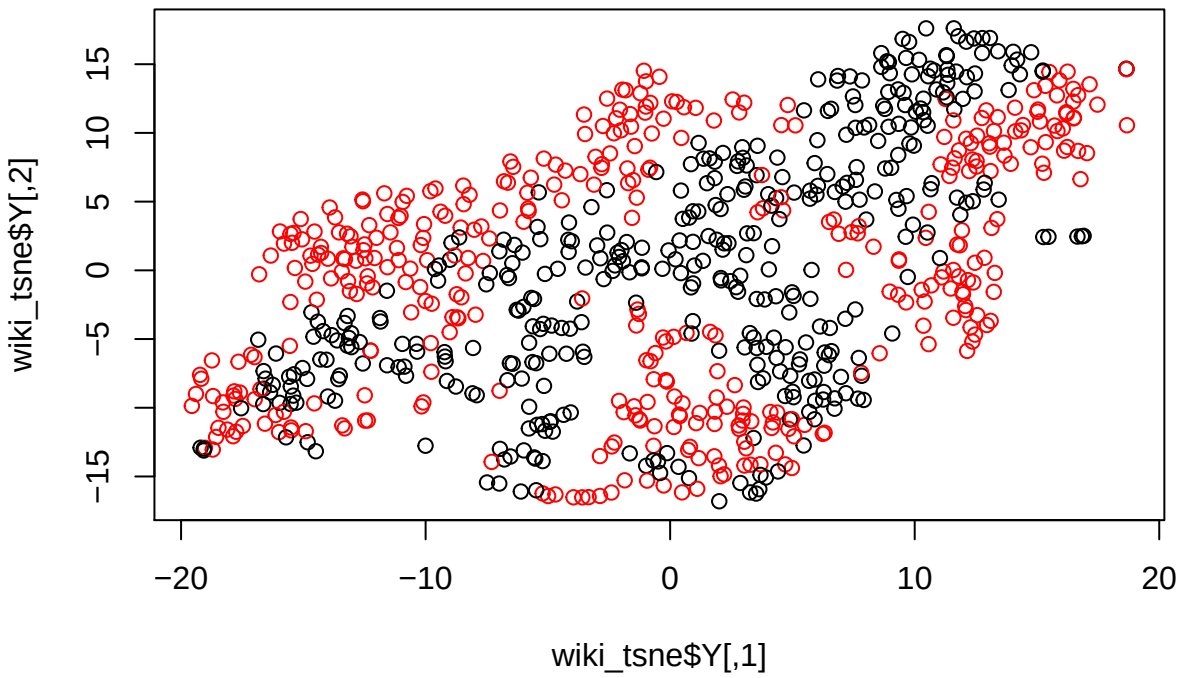
Since we already plotted the first required plot in Q9, I'll just call it below.

```
p1
```



Plot of t-SNE below. As can be seen, the red group tends to have higher absolute value on dimension 1, while the black group tends to have dimension 1 value < 5.

```
plot(wiki_tsne$Y, col = k2$cluster)
```



Coloring the dots by cluster membership does help to make more sense of t-SNE plot, yet PCA still performs better at maximizing intergroup distance. Besides t-SNE results are still harder to explain given my current knowledge level.

Thanks for grading!