Clustering

Introduction

In this task, individuals heard speech tokens from a number of different speakers and freely classified them into groups. Based on previous work, I used hierarchical clustering to examine what natural clusters or groups formed as the result of this free classification.

```
# For reproducibility
set.seed(666)
```

```
library(here)
```

```
## here() starts at /Users/jgeller1/Desktop/clustering_project
```

```
library(tidyverse)  # data manipulation
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)    # clustering algorithms
library(factoextra) # clustering visualization
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(dendextend) # for comparing two dendrograms
```

```
##
## ---------------------
## Welcome to dendextend version 1.14.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(fpc) # kmeans clustering
```

Data Preparation

1. I wrangled the DF so that each row corresponds to each talker and each column corresponds to each participant.
2. I removed missing data.
3. I did not standardize the data.

Agglomerative Hierarchical Clustering

All Participants

I am going to cluster the data using average link clustering. Average link clustering computes all pairwise dissimilarities between the elements, and considers the average of these dissimilarities as the distance between clusters.

1. I calculate the dissimilarity matrix using euclidean distance.
2. I compute the clustering with average link.
3. I plot the cluster solution

```
clust_data <- read_csv(here("data", "class_wide_1.csv")) # read in data
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   speaker = col_character(),
##   `54` = col_character()
## )
```

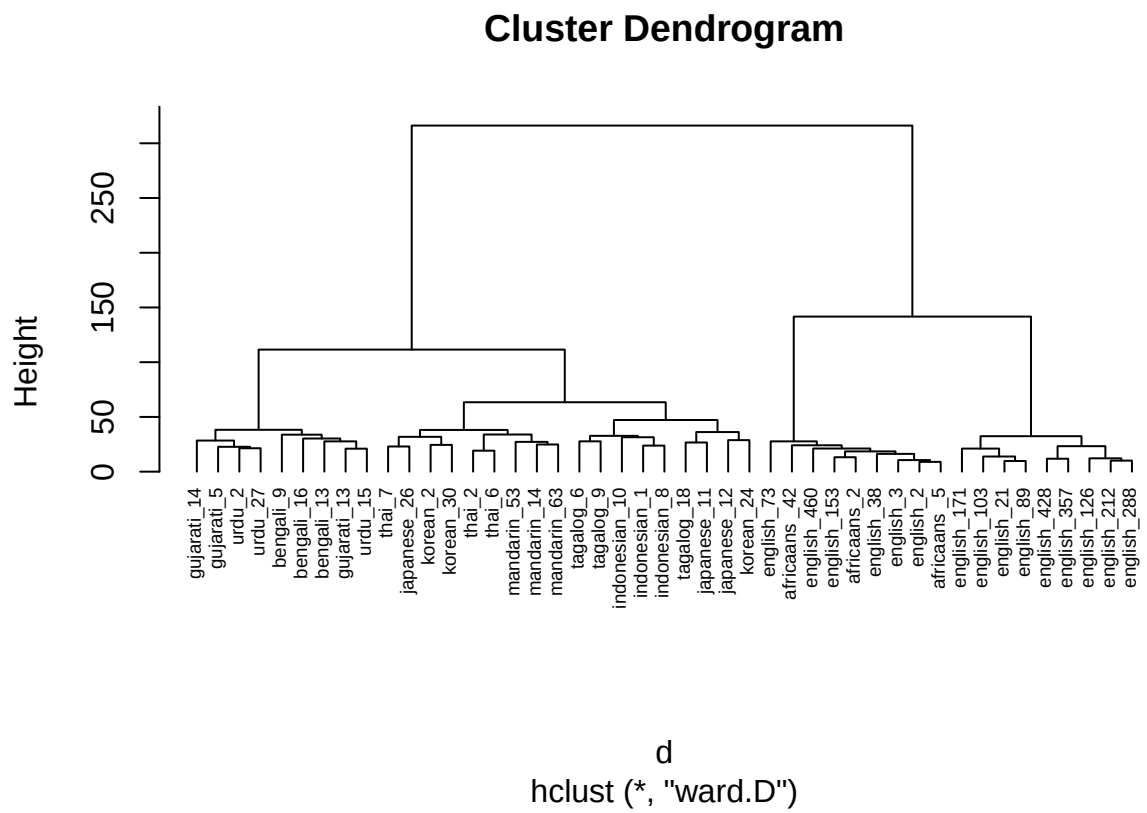
```
## See spec(...) for full column specifications.
```

```
clust_data <- select(clust_data, -X1, -`54`) # remove extra col sub 54 has weird formatting
clust_data <- as.data.frame(clust_data) # turn into df
rownames(clust_data) <- clust_data$speaker # make row names speaker
clust_data <- select(clust_data,-speaker) # remove extra col sub 54 has weird formatting
head(clust_data)# show first couple rows
```

```
##             8 7 1 10 11 12 14 15 16 17 18 19 2 20 23 25 26 27 28 29  3 30 31 32
## bengali_9   1 5 5  1 11  2  1  8  4  2  2  1 9  7  4  5  1  1  1 11  5  1  7  7
## bengali_13  6 5 5  7 14  4  2  7  4  2  6  3 9  1  4  5  4  2  3 11 12  1  8 11
## bengali_16  1 5 5  7  7  4  3  6  2  8  3  3 3  1  3  4  6  1  3 10  2  1  7  8
## gujarati_5  4 5 5  1 14  4  1  7  4  9  9  1 9  4  3  5  4  2  4  8  9  1  7  9
## gujarati_13 1 5 5  1 15  4  2  8  4  2  6  1 9  4  5  5  6  2  4  8  5  1  7  9
## gujarati_14 5 5 5  1  7  4  1  8  4  9  9  3 9  5  7  5  4  4  6  1  5  1  6  9
##             33 34 35 36 38 4 40 41 42 43 44 45 46 47 48 49  5 50 51 52 53 55 56
## bengali_9    9  8 10  8  1 3  1 10  1  1 12  1  5  1  5  8  1  3  7  1  8  9  1
## bengali_13   9  8 10 11  1 4 12  1  1  8 11  1  5  4  1  8  5  4  7  3  8  9  1
## bengali_16   9  8 10 11  6 3  8  8  1  8 11  1  1  2  5  7  5  4  7  3  8  9  8
## gujarati_5   9  6 10  8  1 2  8  7  1  8  8  3 11  2  2  8 10  3  7 11  8  9  7
## gujarati_13  9  8 10  2  1 2 12  1  1  8 11  1 11  6  5  8  2  4  7  1  8  9  7
## gujarati_14  9  6 10  9  1 4 13  2  1  8 11  6  3  4  5  8  3  3  7 15  8  8  8
##             58 59 6 78 87 90 91 96 105 110 111 115 121 123 125 132 133 135 148
## bengali_9    1  5 9  1 11  1 11  1   6  11   1   2   1   9   1   3   1   1   1
## bengali_13   8 11 7  1 11  1 11  1   6  11   7   8   6   9   1   6   2   3   6
## bengali_16   1  6 7  5 11  2  6  1   6  10   6   8   6  10  10   4   2   3   1
## gujarati_5   3 11 7  1 11  2 10  1   1  11   6   8   6   9  10   4   3   2   3
```

```
## gujarati_13 10  5 7  1 11  2 11  1   6  11   2   8   6   9  10   4   2   1   1
## gujarati_14  6  5 6  2 11  8 11  1   6  11   6   9   6   9   8   4   2   1   2
##             151 152 153 155 156 157 158 159 160 161 162 163 164 165 166 167 168
## bengali_9     5   1   8   4   8   6   5   1   5   1   7   4   1   5   1   7   1
## bengali_13    9   1   8   4  10   6   5   2   1  14   7   6   4   5   1   4   2
## bengali_16    7   4   8  11   8   6   3   2   3  14   7   7   1   5   1   7   4
## gujarati_5    5   3   2   5   9   6   3   1   3  14   7   5   6   6   1   4   2
## gujarati_13   5   1   2   5   9   6   3   1   4  14   7   5   6   5   1   7   3
## gujarati_14   5   3   8   5   9   6   3   8   4   5   3   7   6   7   6   7   5
##             169
## bengali_9     2
## bengali_13    4
## bengali_16    4
## gujarati_5    5
## gujarati_13   6
## gujarati_14   1
```

```r
# Dissimilarity matrix
d <- dist(clust_data, method = "euclidean")

# Hierarchical clustering using Average Linkage
hc1 <- hclust(d, method = "ward.D" )

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



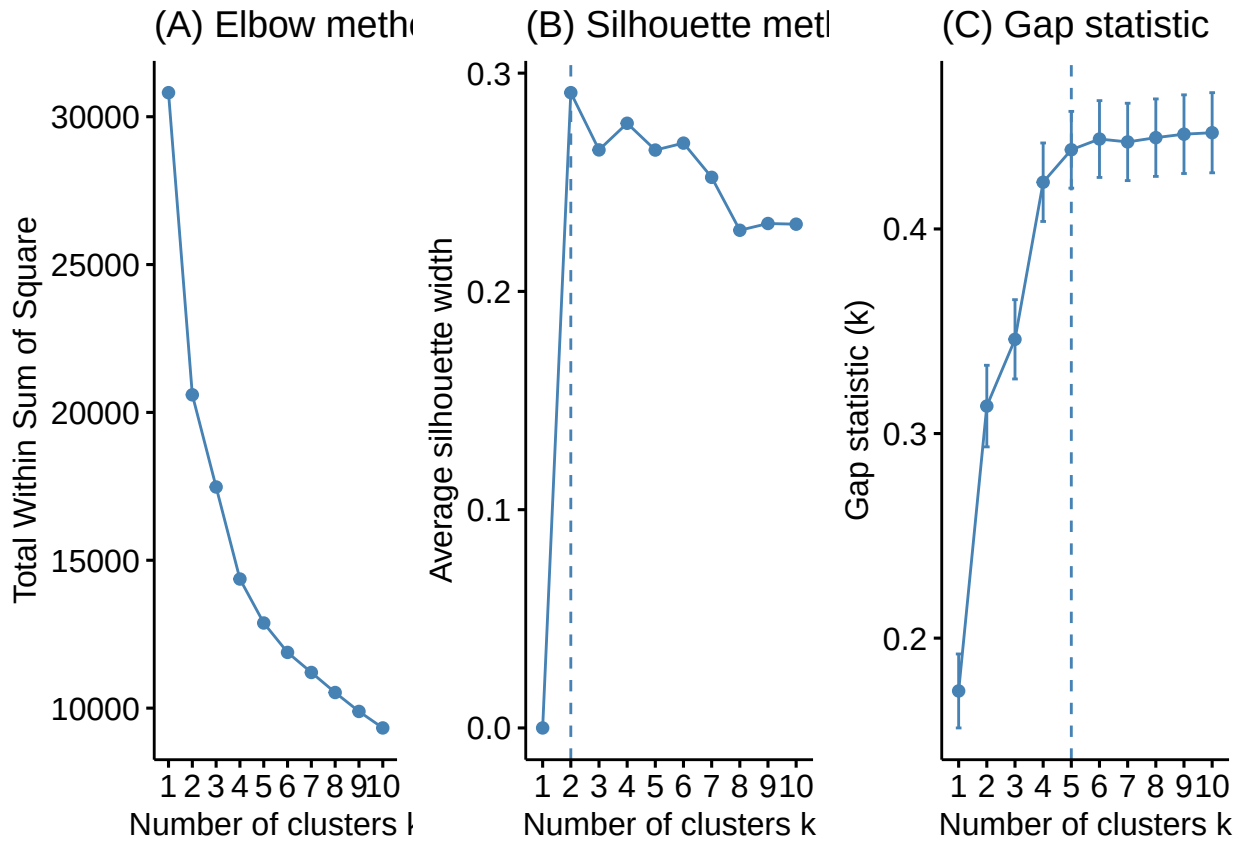
How Many Clusters? In the dendrogram displayed above, each leaf corresponds to one observation. As we move up the tree, observations that are similar to each other are combined into branches, which are themselves fused at a higher height.

The height of the fusion, provided on the vertical axis, indicates the (dis)similarity between two observations. The higher the height of the fusion, the less similar the observations are. Note that, conclusions about the proximity of two observations can be drawn only based on the height where branches containing those two observations first are fused. We cannot use the proximity of two observations along the horizontal axis as a criteria of their similarity.

Although hierarchical clustering provides a fully connected dendrogram representing the cluster relationships, you may still need to choose the preferred number of clusters to extract. Fortunately we can execute approaches similar to k-means clustering. The following compares results provided by the elbow, silhouette, and gap statistic methods. There is no definitively clear optimal number of clusters in this case; although, the silhouette method and Elbow method suggests anywhere between 2-5 clusters.

Humans cant live with this ambiguity. Let's use k-means clustering to determine the number of clusters we should use.

```
# Plot cluster results
p1 <- fviz_nbclust(clust_data, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(clust_data, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(clust_data, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")

# Display plots side by side
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```




```
ggsave("HCstats.png", width=10, height=8)
```

K-means

K-means is another type of clustering algorithm. For a more objective way to determine how many clusters there are, I am going to run k-means clustering over a range of cluster values (here 3-10 clusters). I will use the `fpc` package and the `kmeansrun` function. This function iterates over a number of clusters and chooses the best number of clusters.

```
#run kmeans over a number of ranges (3:10) here

cl <- kmeansruns(clust_data, krange = 4:10, iter.max = 1000)

# pick the best one
cl$bestk
```

```
## [1] 4
```

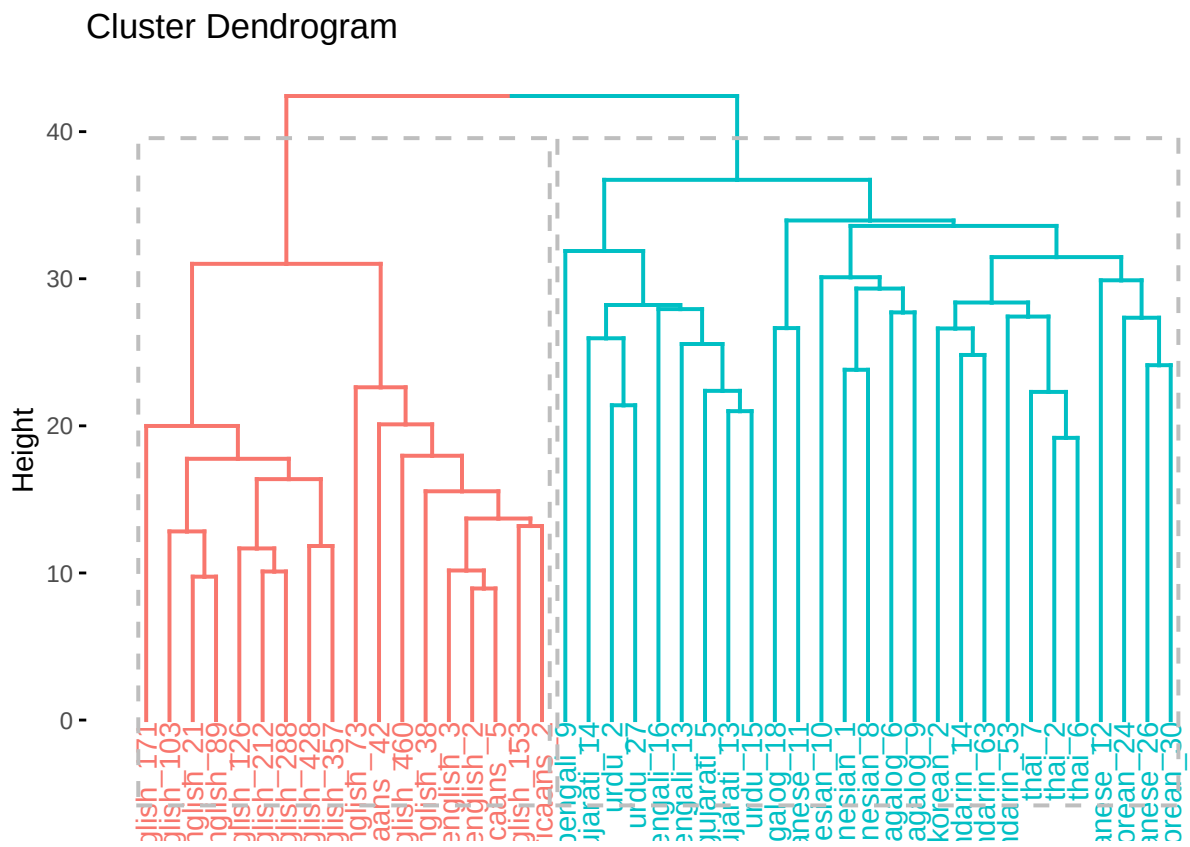
The k-means analysis suggests 2 clusters is best. I personally think 3 clusters better represents the data. It is really a subjective call on your part. Let's visualize both to see what the clusters look like.

Visualize Clusters

Dendogram

2 clusters Here is a dendogram cut at 2.

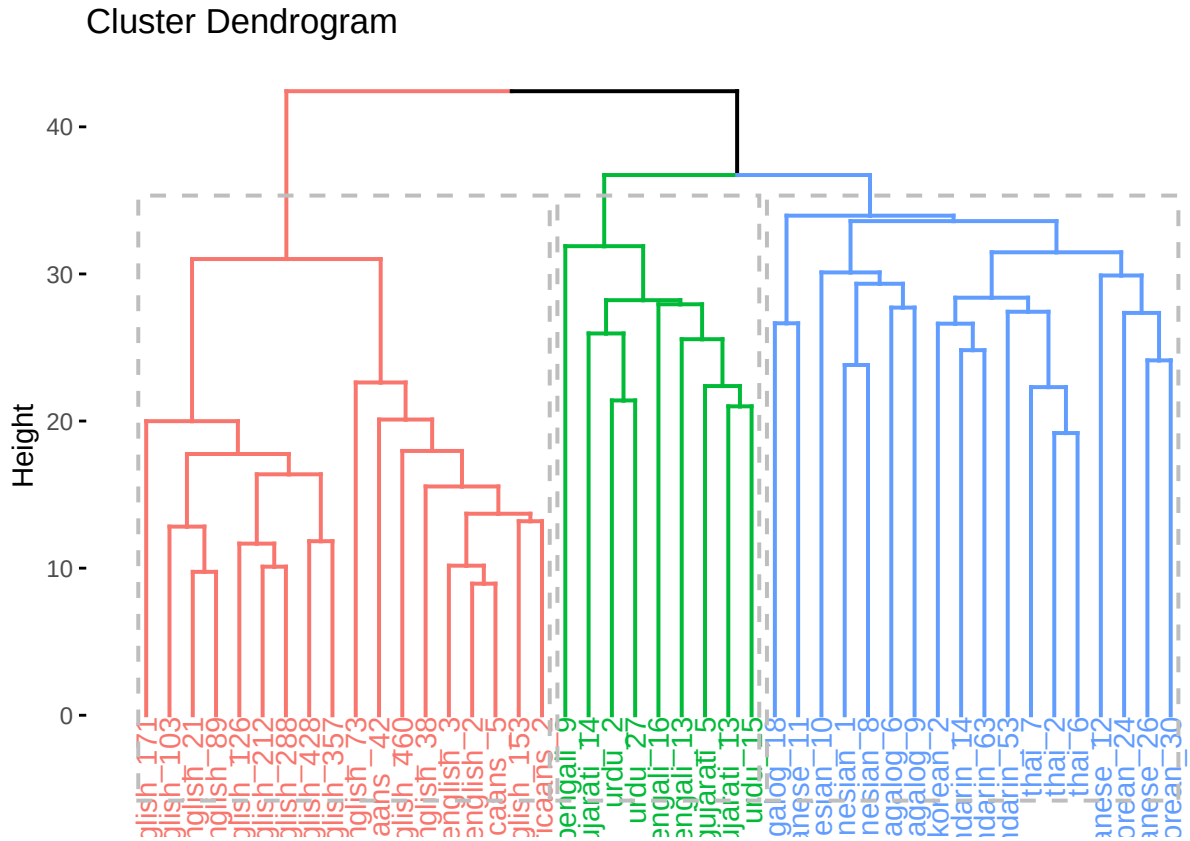
```
hc.cut <- hcut(clust_data, k = 2, hc_method = "average")

fviz_dend(hc.cut, show_labels = TRUE, rect = TRUE)
```




```
ggsave("dendogram2.png", width=10, height=8, dpi=700)
```

Here is a dendogram cut at 3.

```
hc.cut <- hcut(clust_data, k = 3, hc_method = "average")

fviz_dend(hc.cut, show_labels = TRUE, rect = TRUE)
```




```
ggsave("dendogram3.png", width=10, height=8, dpi=700)
```

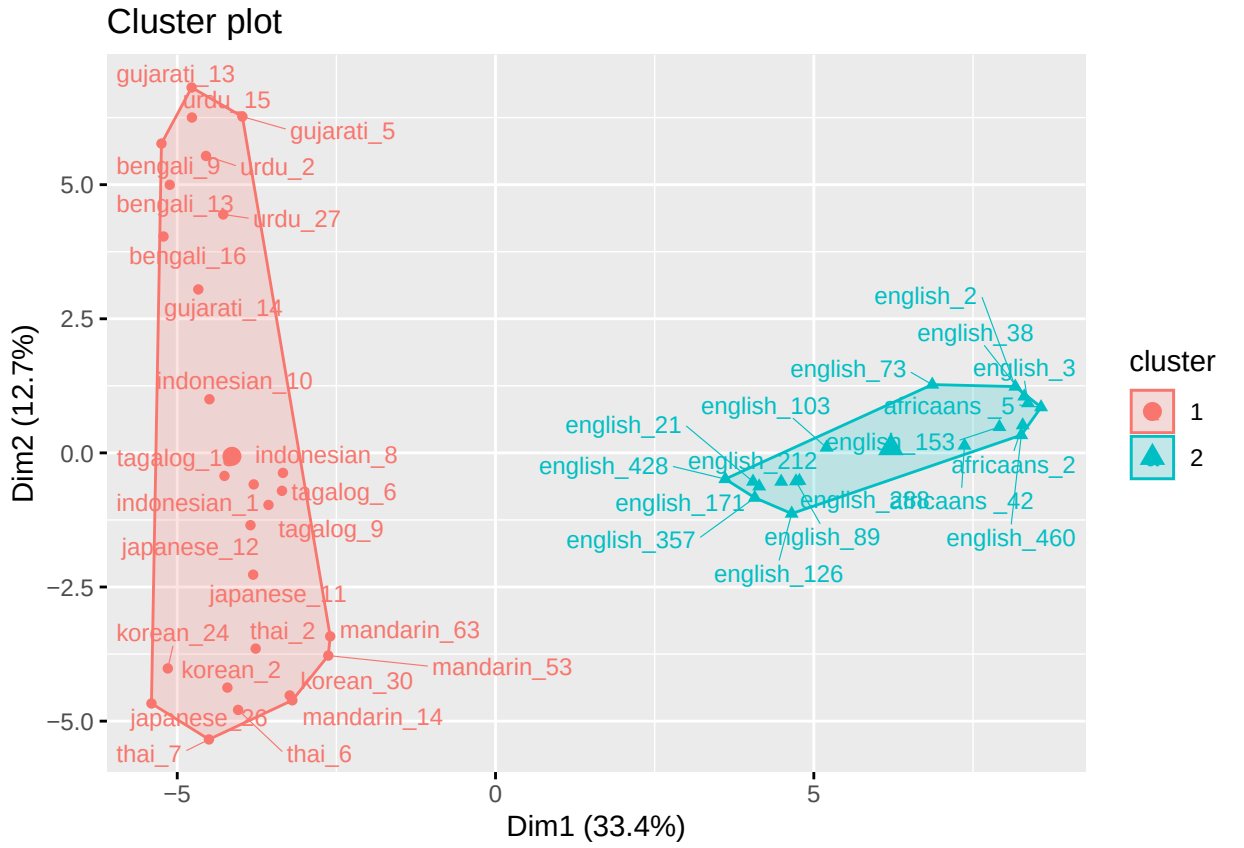
2 Clusters Let's visualize the clusters in two dimensions as it is a bit easier to read that the above dendrogram. I saved this cluster figure as "2clust.png."

```
# Cut tree into 3 groups
sub_grp <- cutree(hc.cut, k = 2)

# Number of members in each cluster
sub_grp
```

```
##     bengali_9    bengali_13    bengali_16    gujarati_5   gujarati_13
##             1             1             1             1             1
##   gujarati_14       urdu_2      urdu_15      urdu_27  indonesian_1
##             1             1             1             1             1
##  indonesian_8 indonesian_10     tagalog_6     tagalog_9    tagalog_18
##             1             1             1             1             1
##        thai_2        thai_6        thai_7   japanese_11   japanese_12
##             1             1             1             1             1
##   japanese_26      korean_2     korean_24     korean_30   mandarin_14
##             1             1             1             1             1
##   mandarin_53   mandarin_63    english_21    english_89   english_103
##             1             1             2             2             2
##   english_428   english_212   english_357   english_288   english_171
##             2             2             2             2             2
##   english_126     english_3    english_73   english_153     english_2
##             2             2             2             2             2
```

```
##    english_38   english_460   africaans_2  africaans _5 africaans _42
##             2             2             2             2             2
```

```
s=fviz_cluster(list(data = clust_data, cluster = sub_grp), labelsize = 10, repel = TRUE)

s
```




```
ggsave("2clust.png", width=10, height=8, dpi=700)
```

3 Clusters Let's visualize the clusters in two dimensions as it is a bit easier to read that the above dendrogram. I saved this cluster figure as "3clust.png." I also saved the data with the cluster number of each speech token as "speech_group.csv." With this you can visualize the clusters how you want.

```
# Cut tree into 3 groups
sub_grp <- cutree(hc.cut, k = 3)

# Number of members in each cluster
sub_grp
```

```
##     bengali_9    bengali_13    bengali_16    gujarati_5   gujarati_13
##             1             1             1             1             1
##   gujarati_14        urdu_2       urdu_15       urdu_27  indonesian_1
##             1             1             1             1             2
##  indonesian_8 indonesian_10     tagalog_6     tagalog_9    tagalog_18
##             2             2             2             2             2
##        thai_2        thai_6        thai_7   japanese_11   japanese_12
##             2             2             2             2             2
##   japanese_26      korean_2     korean_24     korean_30   mandarin_14
```

```
##             2             2             2             2             2
##   mandarin_53   mandarin_63    english_21    english_89   english_103
##             2             2             3             3             3
##   english_428   english_212   english_357   english_288   english_171
##             3             3             3             3             3
##   english_126     english_3    english_73   english_153     english_2
##             3             3             3             3             3
##    english_38   english_460   africaans_2  africaans _5 africaans _42
##             3             3             3             3             3
```

```
s=fviz_cluster(list(data = clust_data, cluster = sub_grp), labelsize = 10, repel = TRUE)

ggsave("3clust.png", width=10, height=8, dpi=700)
```

From this, we glean that two clusters are adequate, at least for all participants.

I think 3 better represents the data, however.

- Cluster 1: English/African
- Cluster 2: Indo/European
- Cluster 3: Asian

Just to summarize, I ran a hierarchical clustering analysis using the average link method to classify talkers in a free classification task. Because there was some ambiguity in terms of the correct correct number of clusters, I ran an iterative k-means analysis ranging from two clusters to ten cluster. This analysis suggested we should use two clusters. If you think three clusters better represents the data please use three instead.

Separate by Language (Mono vs. Multi)

```
clust_data <- read_csv(here("data", "class_wide_1.csv")) # read in data
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   speaker = col_character(),
##   `54` = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
lang_data <- read_csv(here("data", "lang_id.csv"))
```

```
## Parsed with column specification:
## cols(
##   ID = col_double(),
##   language = col_double()
## )
```

```
clust_data <- select(clust_data, -X1, -`54`) # remove extra col sub 54 has weird formatting

clust_data <- clust_data %>% pivot_longer(`8`:`169`) %>%
  rename("ID" = "name")

clust_data$ID<-as.factor(clust_data$ID)
```

```
lang_data$ID<-as.factor(lang_data$ID)

clust_merge <- left_join(clust_data, lang_data)
```

```
## Joining, by = "ID"
```

```
# now sep language
c1_lang_mono <- clust_merge %>%
  filter(language==1) # get

c1_lang_mono <- c1_lang_mono  %>%
  pivot_wider(names_from = ID, values_from = "value")

c1_lang_multi <- clust_merge %>%
  filter(language==0)

c1_lang_multi <- c1_lang_multi  %>%
  pivot_wider(names_from = ID, values_from = "value")

clust_data_mono <- as.data.frame(c1_lang_mono) # turn into df
clust_data_multi <- as.data.frame(c1_lang_multi) # turn into df

rownames(clust_data_mono) <- clust_data_mono$speaker # make row names speaker
rownames(clust_data_multi) <- clust_data_multi$speaker # make row names speaker

clust_data_mono <- select(clust_data_mono,-speaker, -language) # remove extra col sub 54 has weird
clust_data_multi <- select(clust_data_multi,-speaker, -language) # remove extra col sub 54 has weird

head(clust_data_mono)# show first couple rows
```

```
##             8 10 11 14 15 16 17 19 25 28 29 31 33 35 36 40 41 43 44 46 47 48 49
## bengali_9   1  1 11  1  8  4  2  1  5  1 11  7  9 10  8  1 10  1 12  5  1  5  8
## bengali_13  6  7 14  2  7  4  2  3  5  3 11  8  9 10 11 12  1  8 11  5  4  1  8
## bengali_16  1  7  7  3  6  2  8  3  4  3 10  7  9 10 11  8  8  8 11  1  2  5  7
## gujarati_5  4  1 14  1  7  4  9  1  5  4  8  7  9 10  8  8  7  8  8 11  2  2  8
## gujarati_13 1  1 15  2  8  4  2  1  5  4  8  7  9 10  2 12  1  8 11 11  6  5  8
## gujarati_14 5  1  7  1  8  4  9  3  5  6  1  6  9 10  9 13  2  8 11  3  4  5  8
##             51 52 55 56 59 78 90 96 105 123 125 133 135 151 152 153 160 166
## bengali_9    7  1  9  1  5  1  1  1   6   9   1   1   1   5   1   8   5   1
## bengali_13   7  3  9  1 11  1  1  1   6   9   1   2   3   9   1   8   1   1
## bengali_16   7  3  9  8  6  5  2  1   6  10  10   2   3   7   4   8   3   1
## gujarati_5   7 11  9  7 11  1  2  1   1   9  10   3   2   5   3   2   3   1
## gujarati_13  7  1  9  7  5  1  2  1   6   9  10   2   1   5   1   2   4   1
## gujarati_14  7 15  8  8  5  2  8  1   6   9   8   2   1   5   3   8   4   6
```

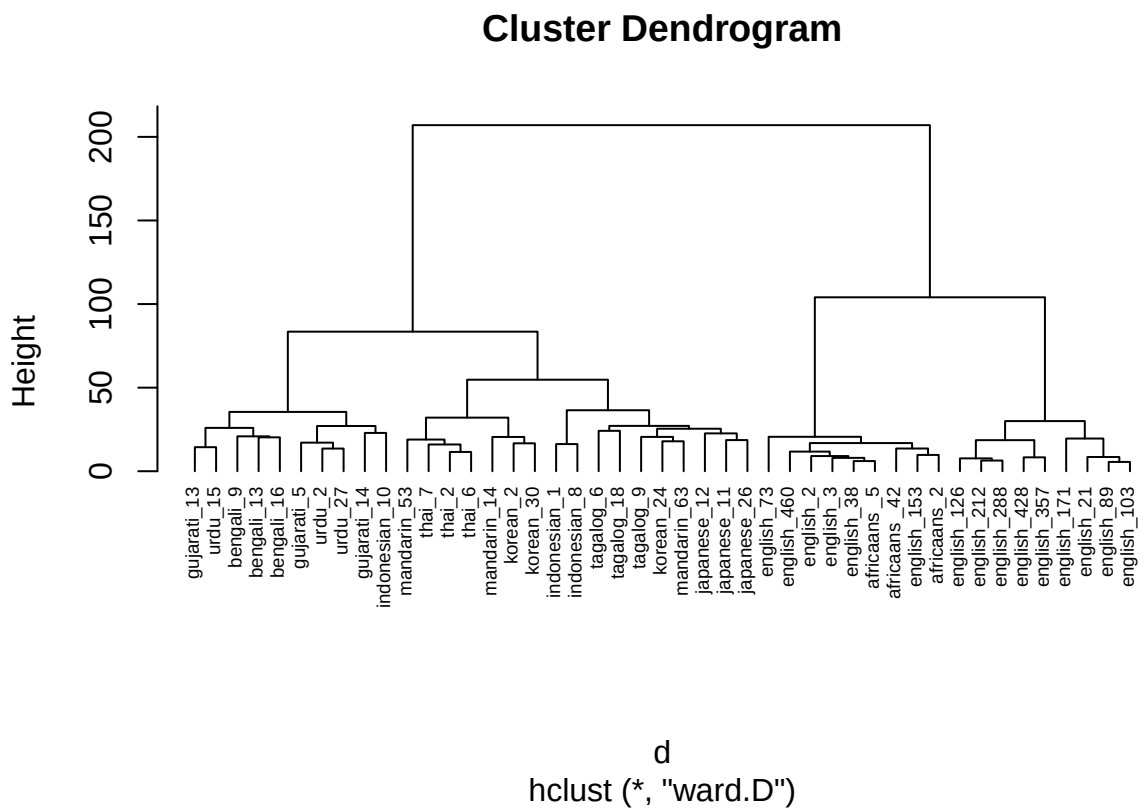
```
head(clust_data_multi)
```

```
##             7 1 12 18 2 20 23 26 27  3 30 32 34 38 4 42 45  5 50 53 58 6 87 91
## bengali_9   5 5  2  2 9  7  4  1  1  5  1  7  8  1 3  1  1  1  3  8  1 9 11 11
## bengali_13  5 5  4  6 9  1  4  4  2 12  1 11  8  1 4  1  1  5  4  8  8 7 11 11
## bengali_16  5 5  4  3 3  1  3  6  1  2  1  8  8  6 3  1  1  5  4  8  1 7 11  6
## gujarati_5  5 5  4  9 9  4  3  4  2  9  1  9  6  1 2  1  3 10  3  8  3 7 11 10
## gujarati_13 5 5  4  6 9  4  5  6  2  5  1  9  8  1 2  1  1  2  4  8 10 7 11 11
## gujarati_14 5 5  4  9 9  5  7  4  4  5  1  9  6  1 4  1  6  3  3  8  6 6 11 11
##             110 111 115 121 132 148 155 156 157 158 159 161 162 163 164 165 167
```

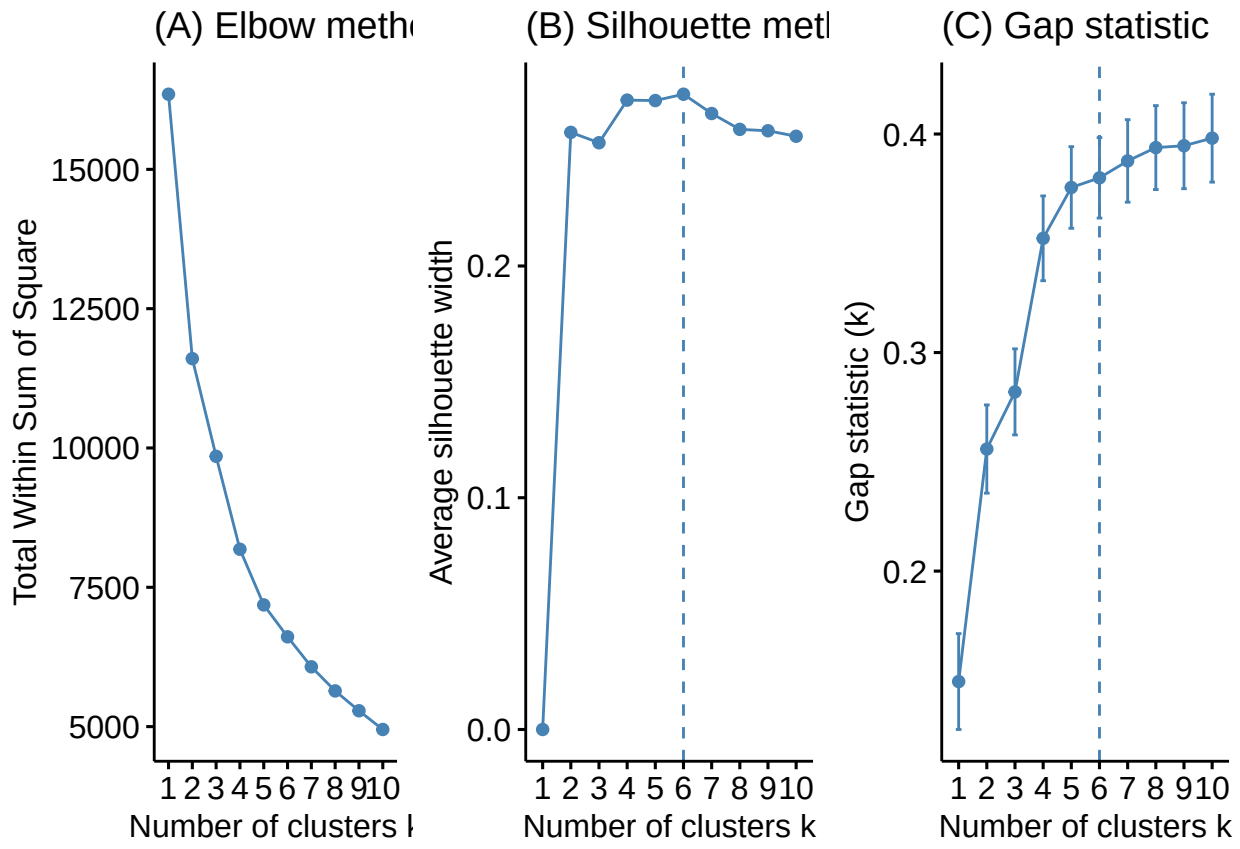
```
## bengali_9    11   1   2   1   3   1   4   8   6   5   1   1   7   4   1   5   7
## bengali_13   11   7   8   6   6   6   4  10   6   5   2  14   7   6   4   5   4
## bengali_16   10   6   8   6   4   1  11   8   6   3   2  14   7   7   1   5   7
## gujarati_5   11   6   8   6   4   3   5   9   6   3   1  14   7   5   6   6   4
## gujarati_13  11   2   8   6   4   1   5   9   6   3   1  14   7   5   6   5   7
## gujarati_14  11   6   9   6   4   2   5   9   6   3   8   5   3   7   6   7   7
##             168 169
## bengali_9     1   2
## bengali_13    2   4
## bengali_16    4   4
## gujarati_5    2   5
## gujarati_13   3   6
## gujarati_14   5   1
```

Monolingual speakers

```
# Dissimilarity matrix
d <- dist(clust_data_mono, method = "euclidean")

# Hierarchical clustering using Average Linkage
hc1 <- hclust(d, method = "ward.D")

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



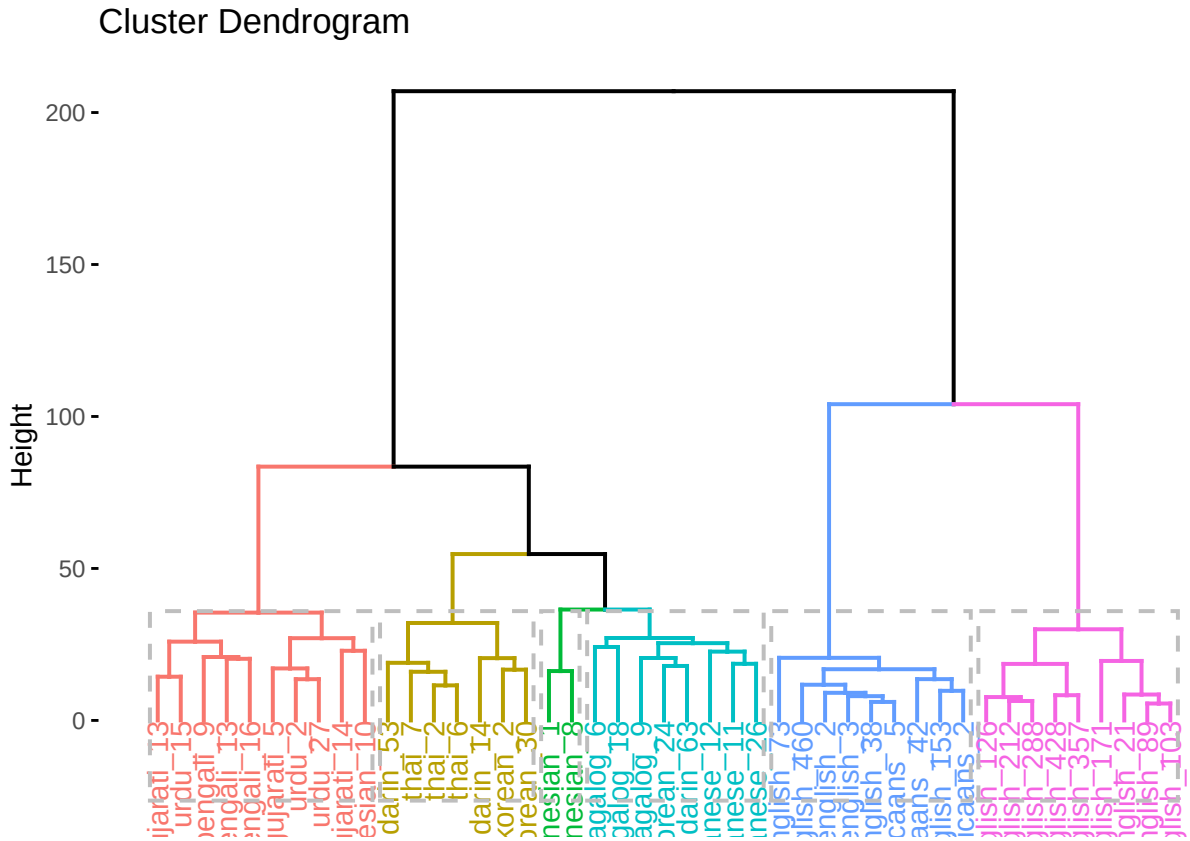
```
# Plot cluster results
p1 <- fviz_nbclust(clust_data_mono, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(clust_data_mono, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(clust_data_mono, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")

# Display plots side by side
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```




```
ggsave("HCstats_mono.png", width=10, height=8)
```

This seems to suggest that 6 clusters might be better.

```
hc.cut <- hcut(clust_data_mono, k = 6, hc_method = "ward.D")

fviz_dend(hc.cut, show_labels = TRUE, rect = TRUE)
```



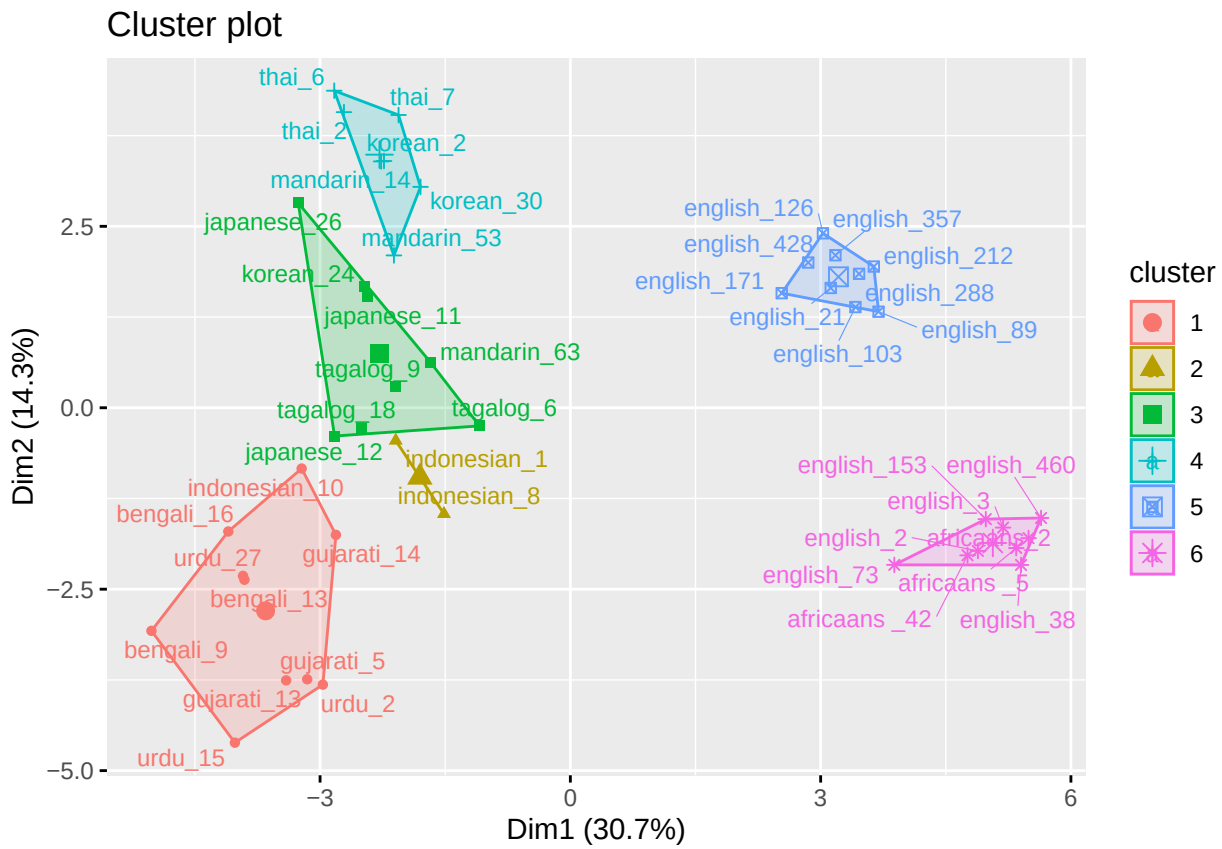
6 clusters

```
ggsave("dendogram6_mono.png", width=10, height=8, dpi=700)
```

```
# Cut tree into 3 groups
sub_grp <- cutree(hc.cut, k = 6)

# Number of members in each cluster
sub_grp
```

```
##     bengali_9    bengali_13    bengali_16    gujarati_5   gujarati_13
##             1             1             1             1             1
##   gujarati_14        urdu_2       urdu_15       urdu_27  indonesian_1
##             1             1             1             1             2
##  indonesian_8 indonesian_10     tagalog_6     tagalog_9    tagalog_18
##             2             1             3             3             3
##        thai_2        thai_6        thai_7   japanese_11   japanese_12
##             4             4             4             3             3
##   japanese_26      korean_2     korean_24     korean_30   mandarin_14
##             3             4             3             4             4
##   mandarin_53   mandarin_63    english_21    english_89   english_103
##             4             3             5             5             5
##   english_428   english_212   english_357   english_288   english_171
##             5             5             5             5             5
##   english_126     english_3    english_73   english_153     english_2
##             5             6             6             6             6
##    english_38   english_460   africaans_2  africaans _5 africaans _42
##             6             6             6             6             6
```

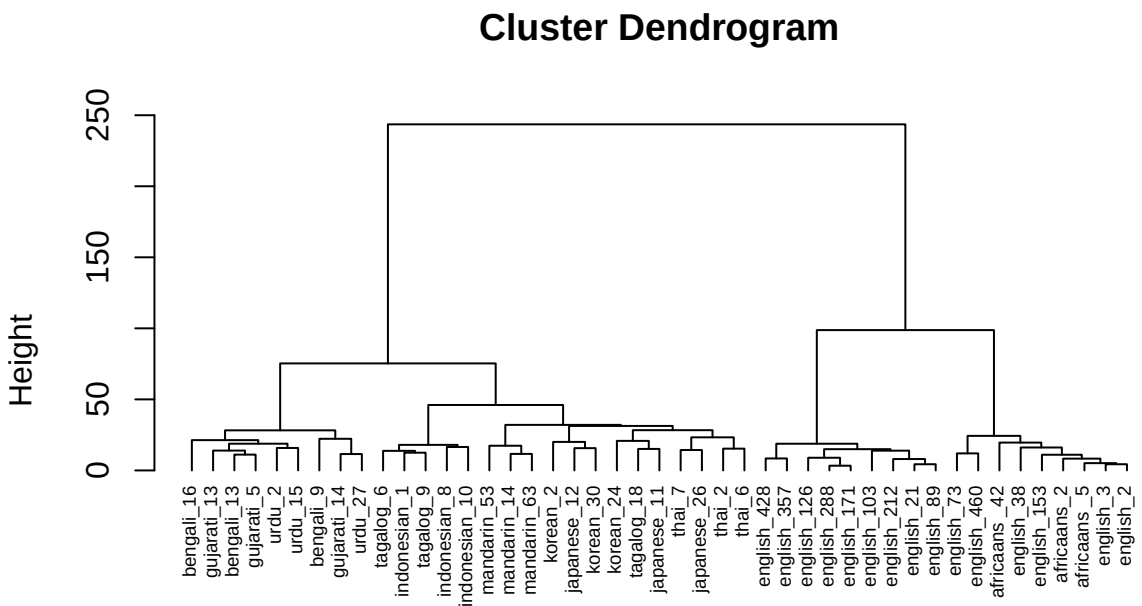
```
s=fviz_cluster(list(data = clust_data_mono, cluster = sub_grp), labelsize = 10, repel = TRUE)

s
```



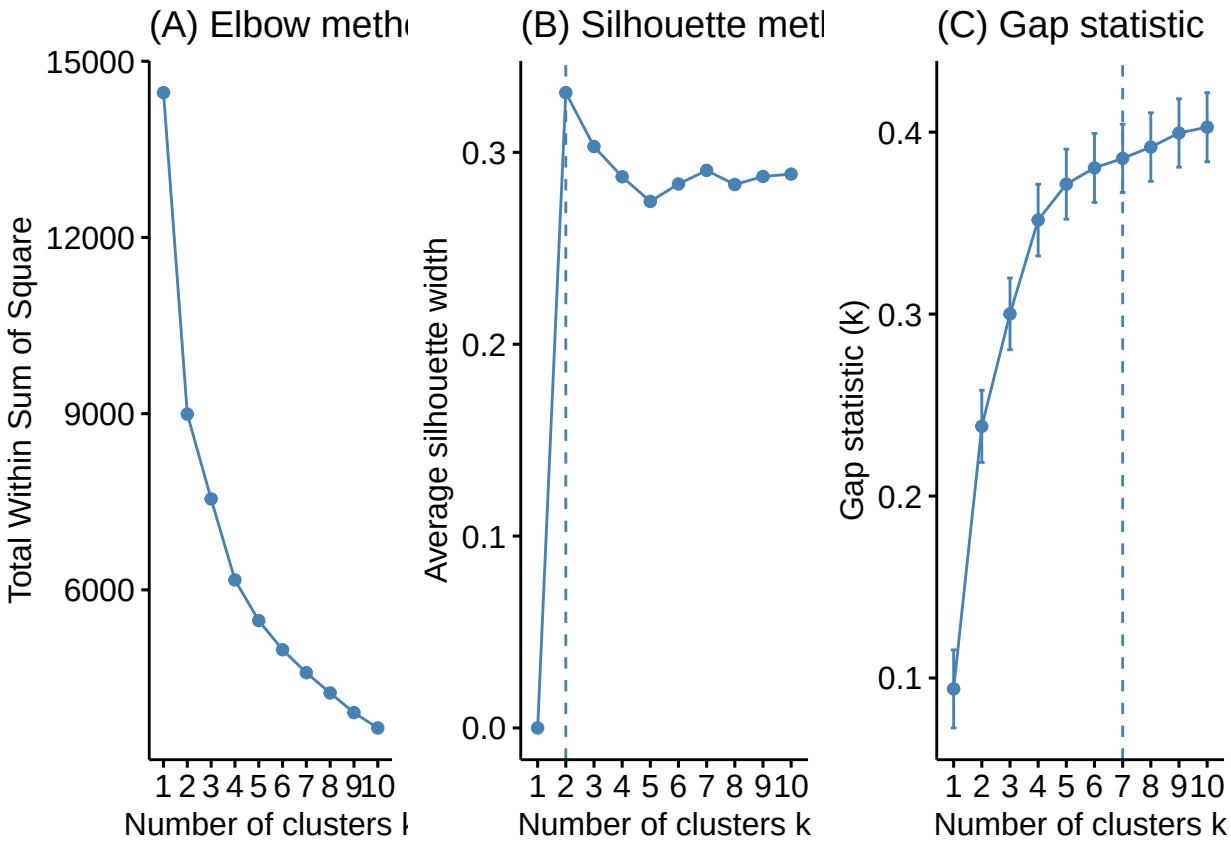

```
ggsave("6clust_mono.png", width=10, height=8, dpi=700)
```

Multilinguial speakers

```
# Dissimilarity matrix
d <- dist(clust_data_multi, method = "euclidean")

# Hierarchical clustering using Average Linkage
hc1 <- hclust(d, method = "ward.D")

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



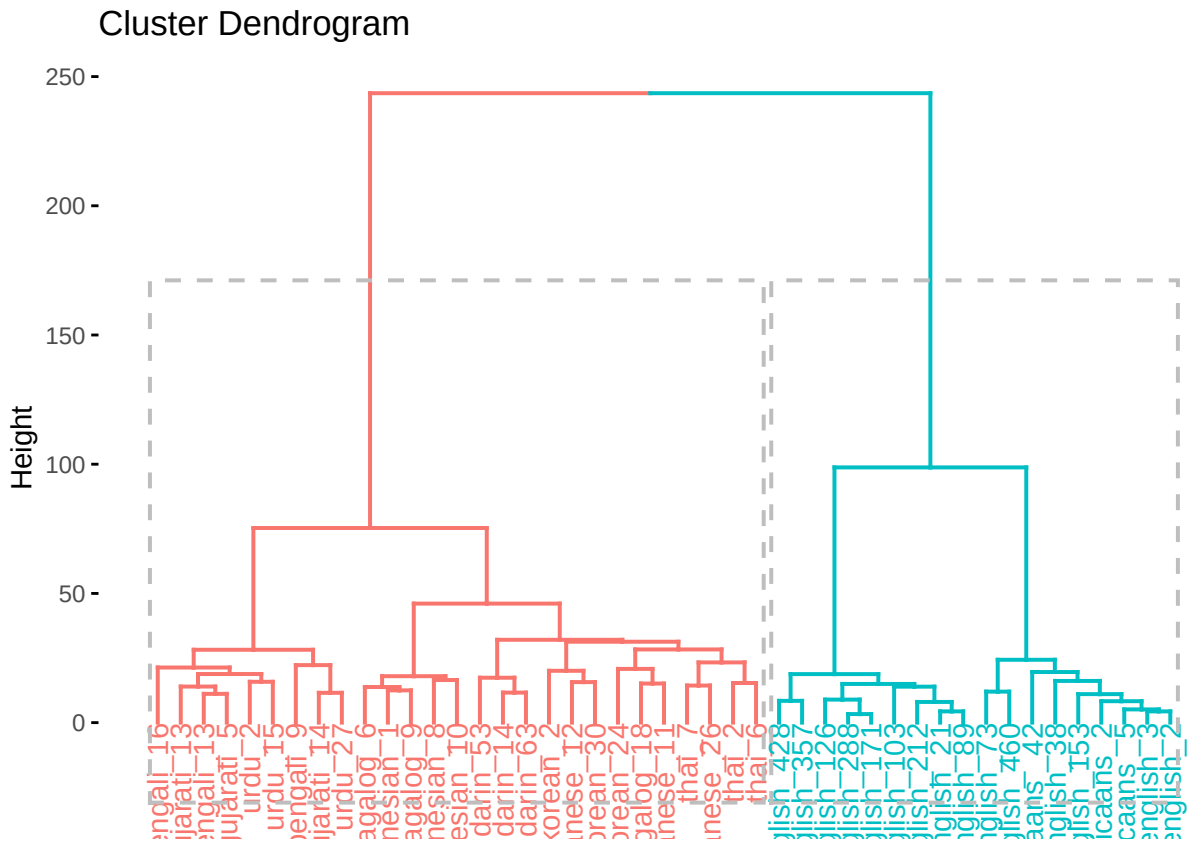

```
# Plot cluster results
p1 <- fviz_nbclust(clust_data_multi, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(clust_data_multi, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(clust_data_multi, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")

# Display plots side by side
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```




```
ggsave("HCstats_multi.png", width=10, height=8)
```

Look like 2 clusters fits the data better.

```
hc.cut <- hcut(clust_data_multi, k = 2, hc_method = "ward.D")

fviz_dend(hc.cut, show_labels = TRUE, rect = TRUE)
```



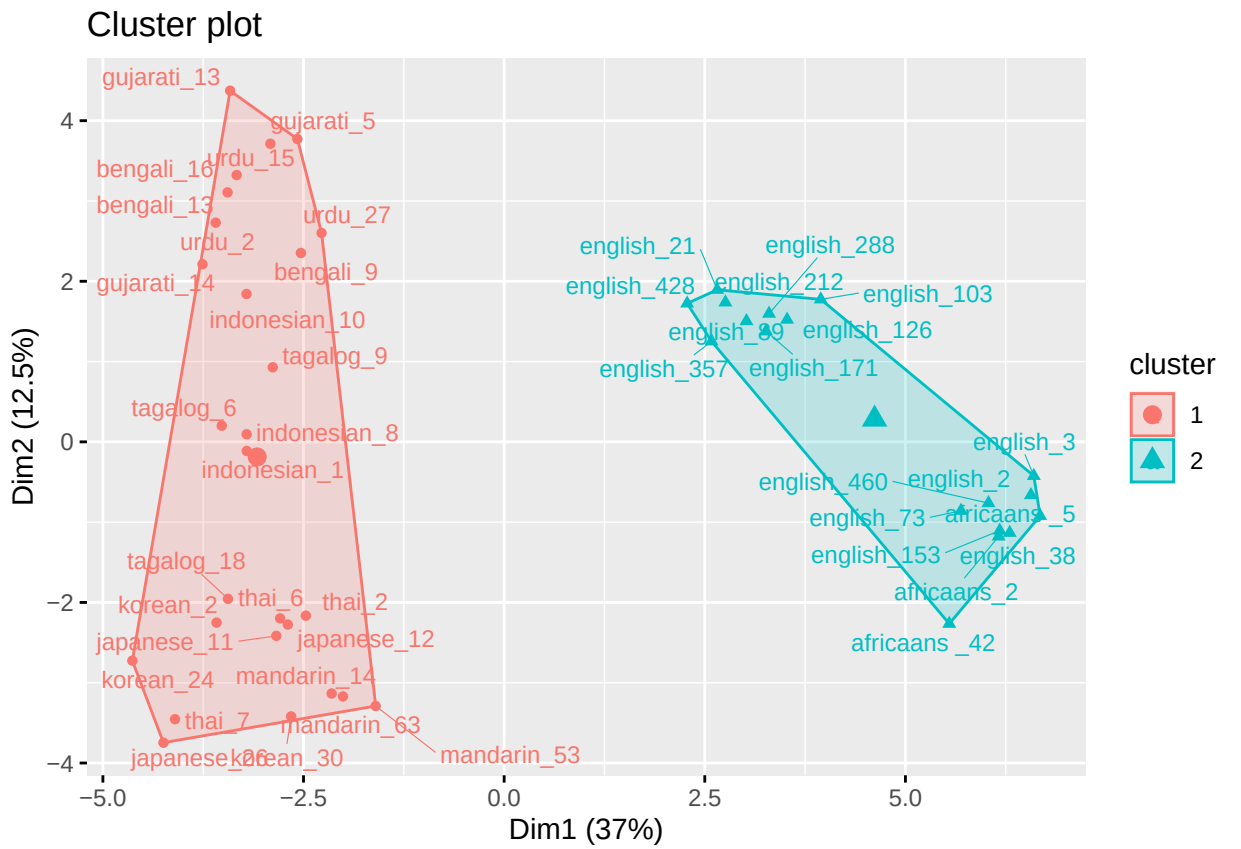
2 clusters

```
ggsave("dendogram2_multi.png", width=10, height=8, dpi=700)
```

```
# Cut tree into 3 groups
sub_grp <- cutree(hc.cut, k = 2)

# Number of members in each cluster
sub_grp
```

```
##     bengali_9    bengali_13    bengali_16    gujarati_5   gujarati_13
##             1             1             1             1             1
##   gujarati_14       urdu_2      urdu_15      urdu_27  indonesian_1
##             1             1             1             1             1
##  indonesian_8 indonesian_10     tagalog_6     tagalog_9    tagalog_18
##             1             1             1             1             1
##        thai_2        thai_6        thai_7   japanese_11   japanese_12
##             1             1             1             1             1
##   japanese_26      korean_2     korean_24     korean_30   mandarin_14
##             1             1             1             1             1
##   mandarin_53   mandarin_63    english_21    english_89   english_103
##             1             1             2             2             2
##   english_428   english_212   english_357   english_288   english_171
##             2             2             2             2             2
##   english_126     english_3    english_73   english_153     english_2
##             2             2             2             2             2
##    english_38   english_460   africaans_2  africaans _5 africaans _42
##             2             2             2             2             2
```

```
s=fviz_cluster(list(data = clust_data_multi, cluster = sub_grp), labelsize = 10, repel = TRUE)

s
```




```
ggsave("2clust_multi.png", width=10, height=8, dpi=700)
```