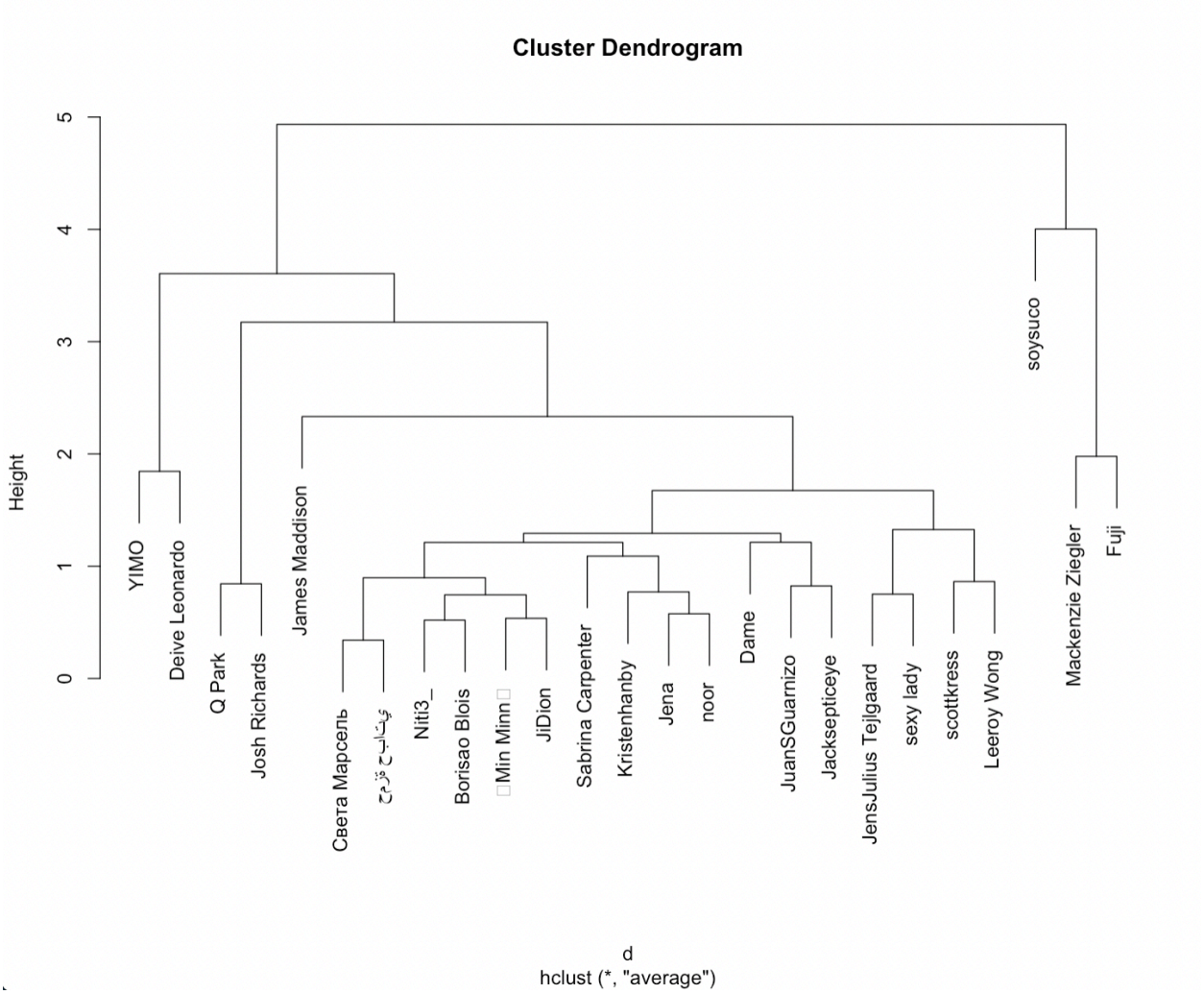
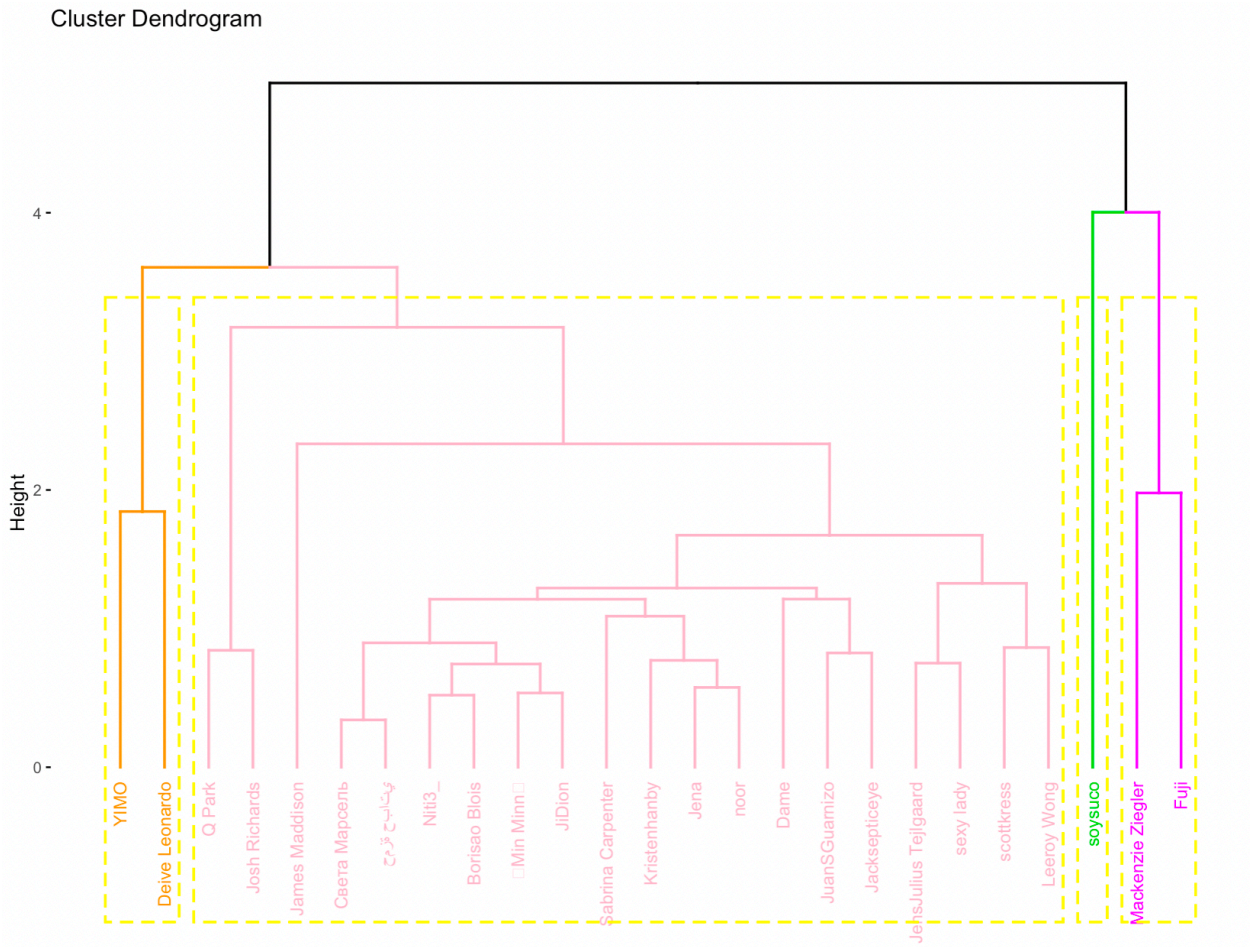
```r
# AD699
# Assignment 5
# Tiange Chang

library(tidyverse)
library(factoextra)
library(ggsci)
library(ggthemes)

# Task 1: Hierarchical Clustering
# 1.
setwd('/Users/t/Desktop')
tt <- read.csv('tiktok_top_1000.csv')
View(tt)

# 2.
dim(tt)
# The dataset's dimensions are 1000 rows by 11 columns.

# 3.
set.seed(25)
tt2 <- tt[sample(nrow(tt), 25), ]
View(tt2)

# 4.
View(tt2)
# By reading the dataset description, the data should be scaled.
# Data ranges are not the same; scaling of the data makes it easy for a model
# to learn and understand the problem.

# normalize input variables
tt.norm <- sapply(tt2[, 6:10], scale)
# set row names
row.names(tt.norm) <- tt2[, 4]
View(tt.norm)

# 5. a
# compute Euclidean distance
d <- dist(tt.norm, method = "euclidean")
d

hc <- hclust(d, method = "average")
plot(hc)
```




```r
# CEX = character expansion
# LWD = line width
fviz_dend(hc, cex = 0.7, lwd = 0.7, k = 4,
          # Manually selected colors
          k_colors = c("orange", "pink", "green3", "magenta"),
          rect = TRUE, rect_border = "yellow", rect_fill = FALSE)
```



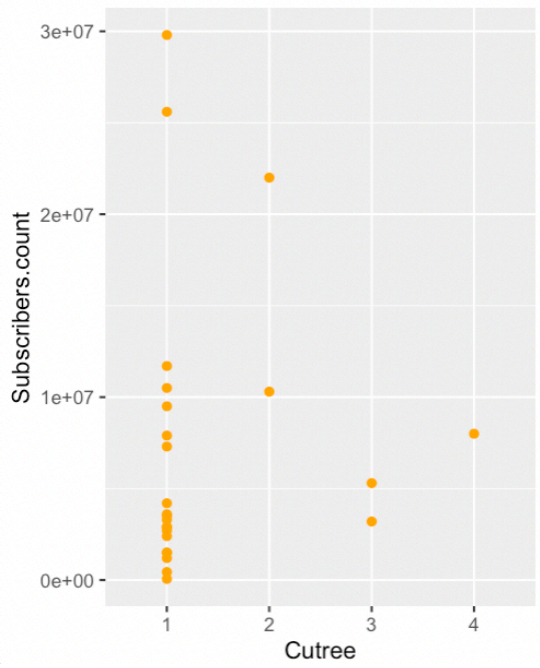

```
# 5. b
# I see four clusters, ordered by four different colors.

# 5. c
# The desired number is 4.
Cutree <- cutree(hc, k = 4, h = 5)
Cutree
# This cut follows the four different color clustering from above.
```

```
> Cutree
             Q Park                Dame JensJulius Tejlgaard        JuanSGuarnizo
                  1                   1                   1                    1
       ✨Min Minn✨    Mackenzie Ziegler                Fuji                YIMO
                  1                   2                   2                    3
      Jacksepticeye             soysuco    Sabrina Carpenter           scottkress
                  1                   4                   1                    1
       Leeroy Wong       Света Марсель               Niti3_       Josh Richards
                  1                   1                   1                    1
      Borisao Blois               Jena                noor        Kristenhanby
                  1                   1                   1                    1
     Deive Leonardo             JiDion           sexy lady          حمزة حباتي
                  3                   1                   1                    1
     James Maddison
                  1
```

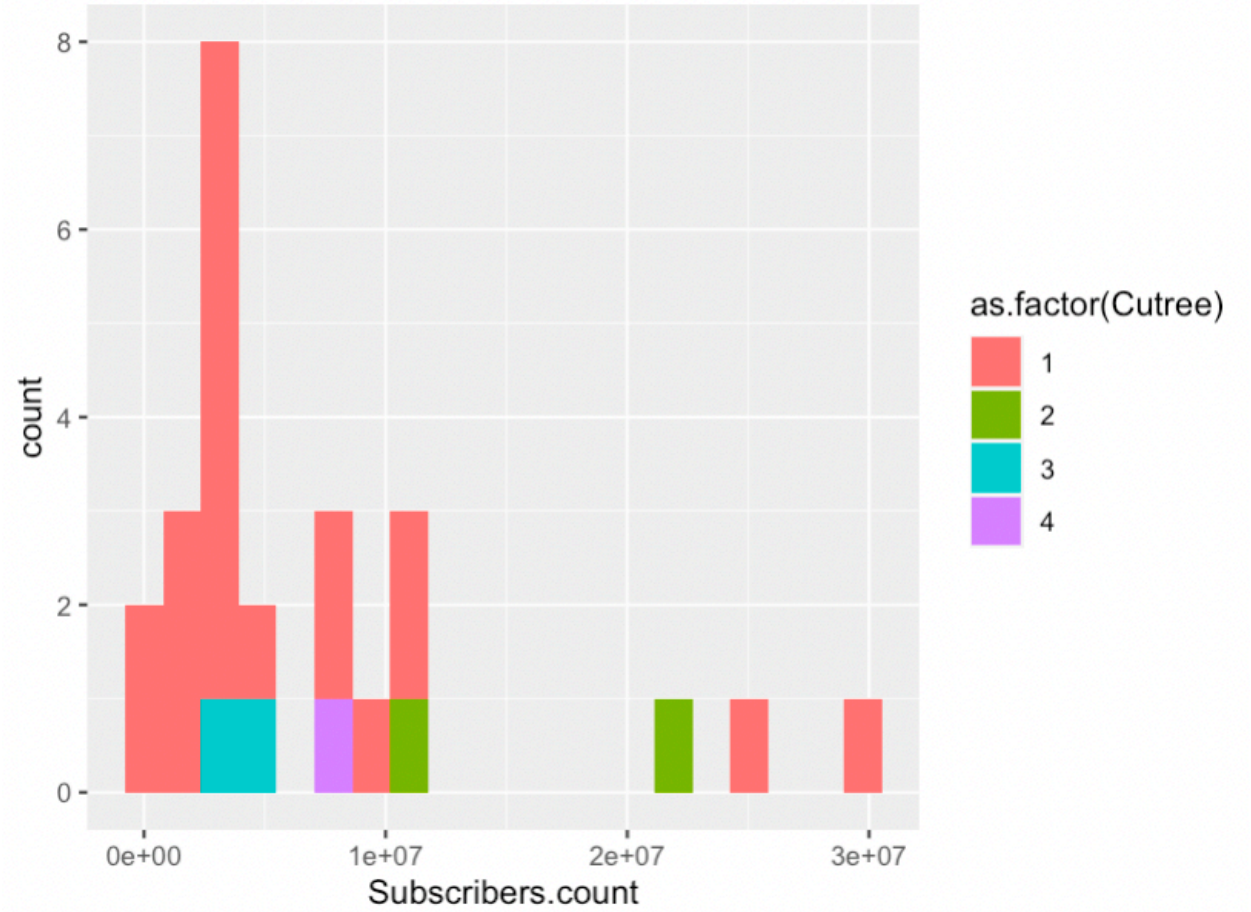
```r
# 5. d
tt2 <- as.data.frame(tt2)
View(tt2)
library(dplyr)
library(ggplot2)
?summarise
tt2$Cutree <- paste(Cutree)
tt2 %>% group_by(Cutree) %>% summarise(Count = n(),
                                       Sub_cnt = mean(Subscribers.count),
                                       View_avg = mean(Views.avg.),
                                       Like_avg = mean(Likes.avg.),
                                       Comment_avg = mean(Comments.avg.),
                                       Share_avg = mean(Shares.avg.))
```

```
# A tibble: 4 × 7
  Cutree Count  Sub_cnt View_avg Like_avg Comment_avg Share_avg
  <chr>  <int>    <dbl>    <dbl>    <dbl>       <dbl>     <dbl>
1 1         20  6625000  2441715   316805       3436.     2473.
2 2          2 16150000  9800000  1400000       6350      3300
3 3          2  4250000  1026550   126200        988.    16000
4 4          1  8000000  6200000   951700      20200      4000
>
```

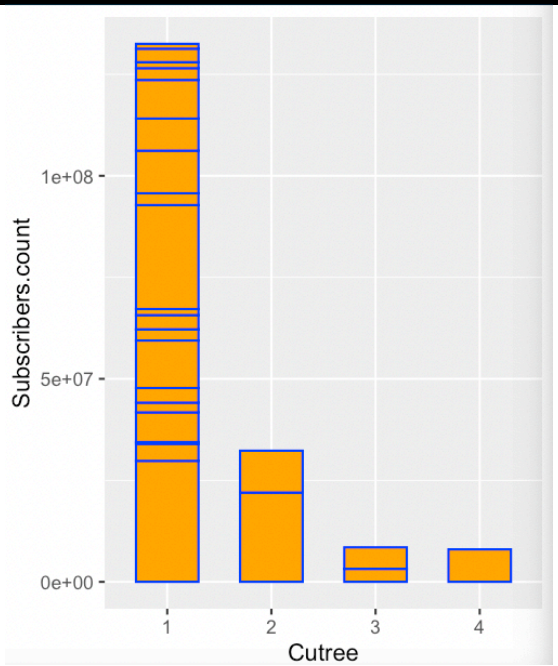
```r
# By running the group by and summarize function, I listed the mean value of
# each cluster. Cluster 2 has the highest subscribers count, view average, likes
# average, and comment average. Cluster 1 has the most artists.

# 5. e
# geom point
ggplot(tt2, aes(Cutree, Subscribers.count)) +
  geom_point(color = "orange")
```



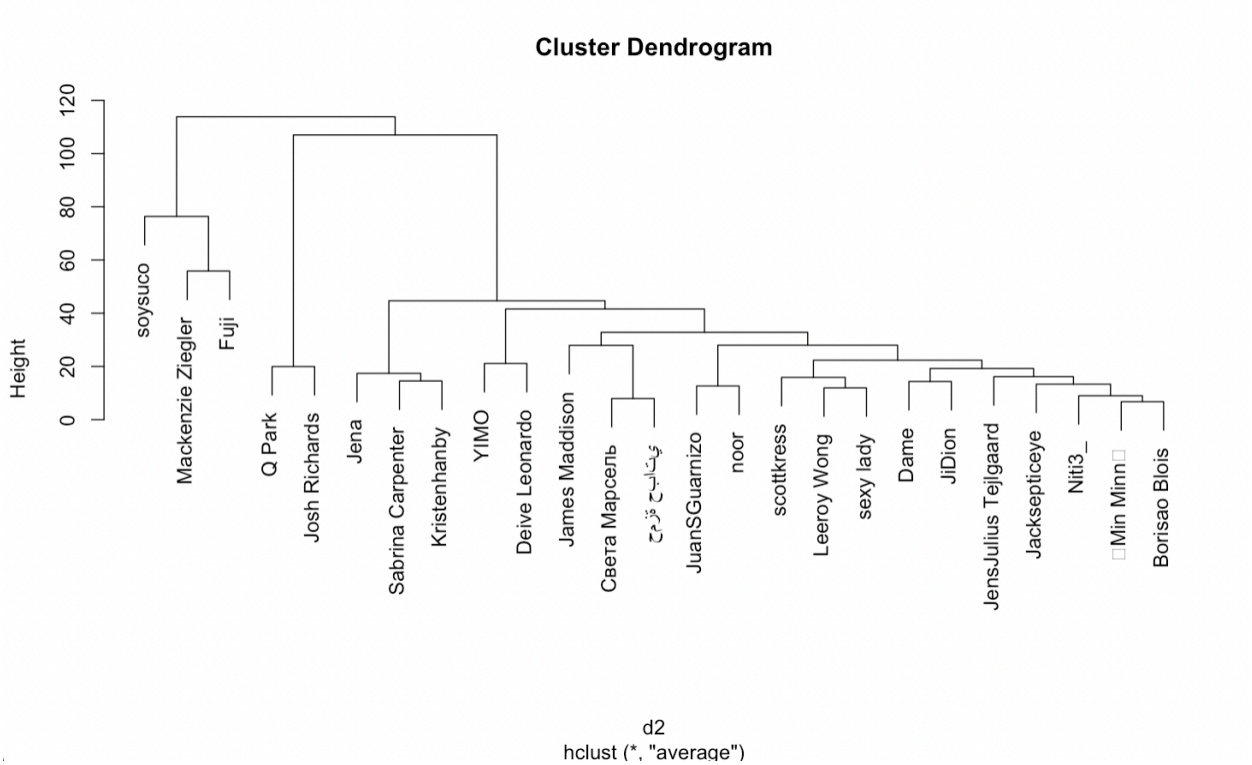
```r
# geom histogram
ggplot(tt2, aes(Subscribers.count, fill = as.factor(Cutree))) +
  geom_histogram(bins = 20)
```



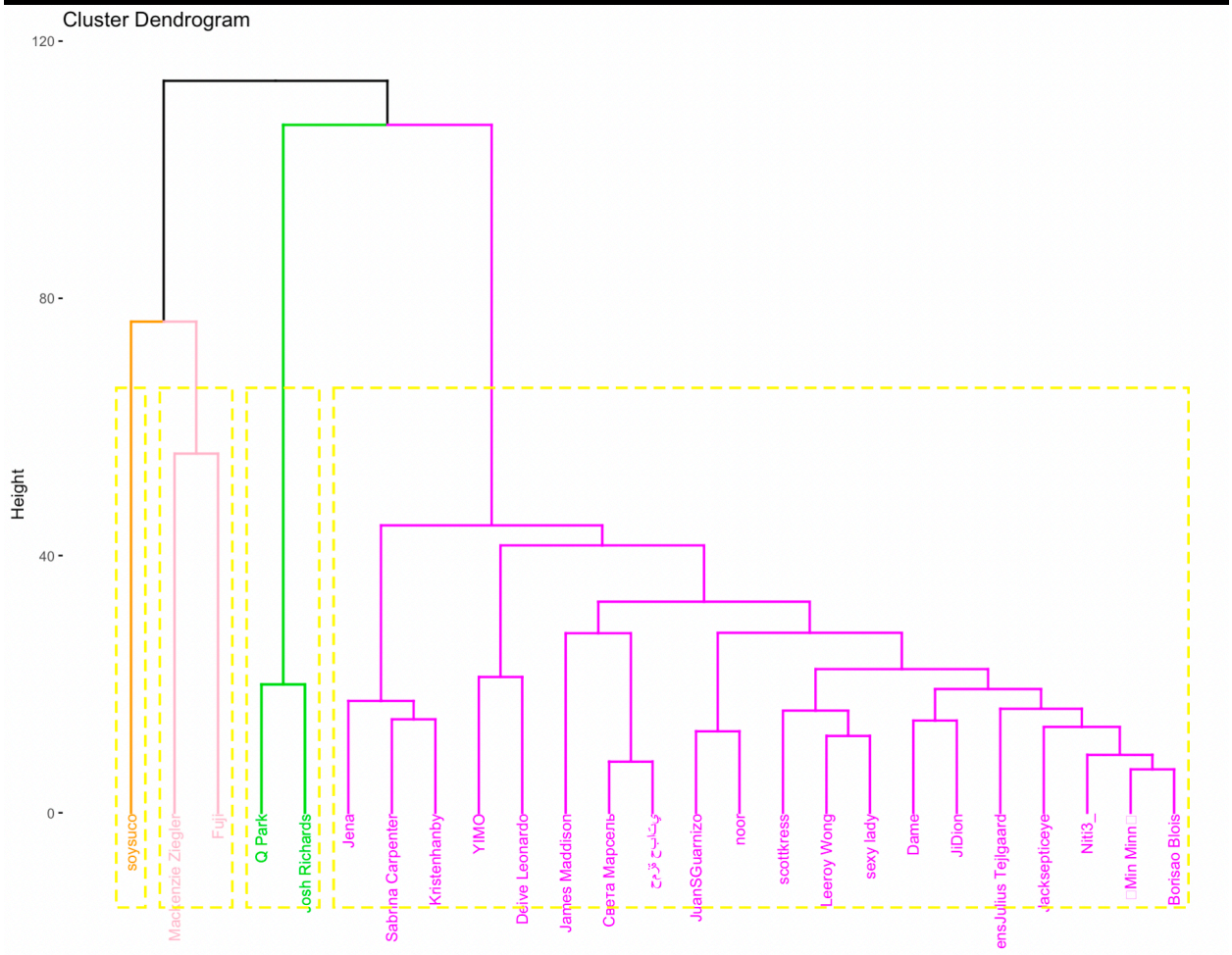

```r
# geom bar
ggplot(tt2, aes(Cutree, Subscribers.count)) +
  geom_bar(width = 0.6, stat = "identity", color = "blue", fill = "orange")
```



```r
# 5. f
# The artist I pick is Fuji, and it falls into cluster 2. There is one more
# artist in this cluster: Mackenzie Ziegler. Their rank is 49 and 72 which is
# the top three in my 25 pick. They both have over 10,000,000 subscribers.
# The rank 59 artist also has high rank but not fall into the same cluster, the
# reason could be less than 10,000,000 subscribers.

# 6.
# It might be problematic to view these variables with equal weight is because
# it is meaningless if five variables have the same weight, the R studio does
# not know the business priorities. So if we want to standardizing variables
# with weighted data, we have to come up with our own weight system.

# 7.
ttw <- as.data.frame(tt.norm)
View(ttw)
ttw$Subscribers.count <- ttw$Subscribers.count * 35
ttw$Views.avg. <- ttw$Views.avg. * 35
ttw$Likes.avg. <- ttw$Likes.avg. * 10
ttw$Comments.avg. <- ttw$Comments.avg. * 10
ttw$Shares.avg. <- ttw$Shares.avg. * 10
View(ttw)

# 7. a
# I think the subscribers count and the views are the most important, and the
# other three are less important. So I used a 100% scale with the two most
# important data have 35%, and the other three with 10%.

# 8. a
# compute Euclidean distance
d2 <- dist(ttw, method = "euclidean")
d2

hc2 <- hclust(d2, method = "average")
plot(hc2)
```


Cluster Dendrogram
Height
0
20
40
60
80
100
120
soysuco
Mackenzie Ziegler
Fuji
Q Park
Josh Richards
Jena
Sabrina Carpenter
Kristenhanby
YIMO
Deive Leonardo
James Maddison
Света Марсель
JuanSGuarnizo
noor
scottkress
Leeroy Wong
sexy lady
Dame
JiDion
JensJulius Tejlgaard
Jacksepticeye
Niti3_
Borisao Blois
d2
hclust (*, "average")

```
# CEX = character expansion
# LWD = line width
fviz_dend(hc2, cex = 0.7, lwd = 0.7, k = 4,
          # Manually selected colors
          k_colors = c("orange", "pink", "green3", "magenta"),
          rect = TRUE, rect_border = "yellow", rect_fill = FALSE)
```



```r
# The new dendrogram still has four clusters, but with different artists and
# orders. Soysuco, Mackenzie Ziegler and Fuji are still in the same cluster but
# the other two has changed. YIMO and Deive Leonardo was in one cluster, but now
# Q park and Josh Richards replaced them.

# 8. b
# The desired number is 4.
Cutree2 <- cutree(hc2, k = 4, h = 90)
Cutree2
# This cut follows the four different color clustering from above.
```

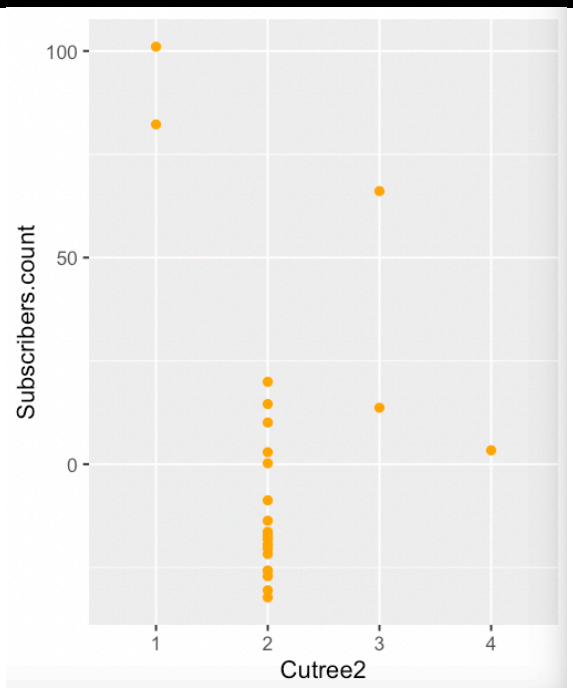
```
> Cutree2
             Q Park                Dame JensJulius Tejlgaard        JuanSGuarnizo
                  1                    2                    2                    2
       ✨Min Minn✨    Mackenzie Ziegler                 Fuji                YIMO
                  2                    3                    3                    2
      Jacksepticeye              soysuco    Sabrina Carpenter           scottkress
                  2                    4                    2                    2
        Leeroy Wong        Света Марсель               Niti3_        Josh Richards
                  2                    2                    2                    1
      Borisao Blois                 Jena                 noor         Kristenhanby
                  2                    2                    2                    2
     Deive Leonardo              JiDion            sexy lady          حمزة حباتي
                  2                    2                    2                    2
     James Maddison
                  2
```

```r
# 8. c
ttw$Cutree2 <- paste(Cutree2)
ttw %>% group_by(Cutree2) %>% summarise(Count = n(),
                                        Sub_cnt = mean(Subscribers.count),
                                        View_avg = mean(Views.avg.),
                                        Like_avg = mean(Likes.avg.),
                                        Comment_avg = mean(Comments.avg.),
                                        Share_avg = mean(Shares.avg.))
```

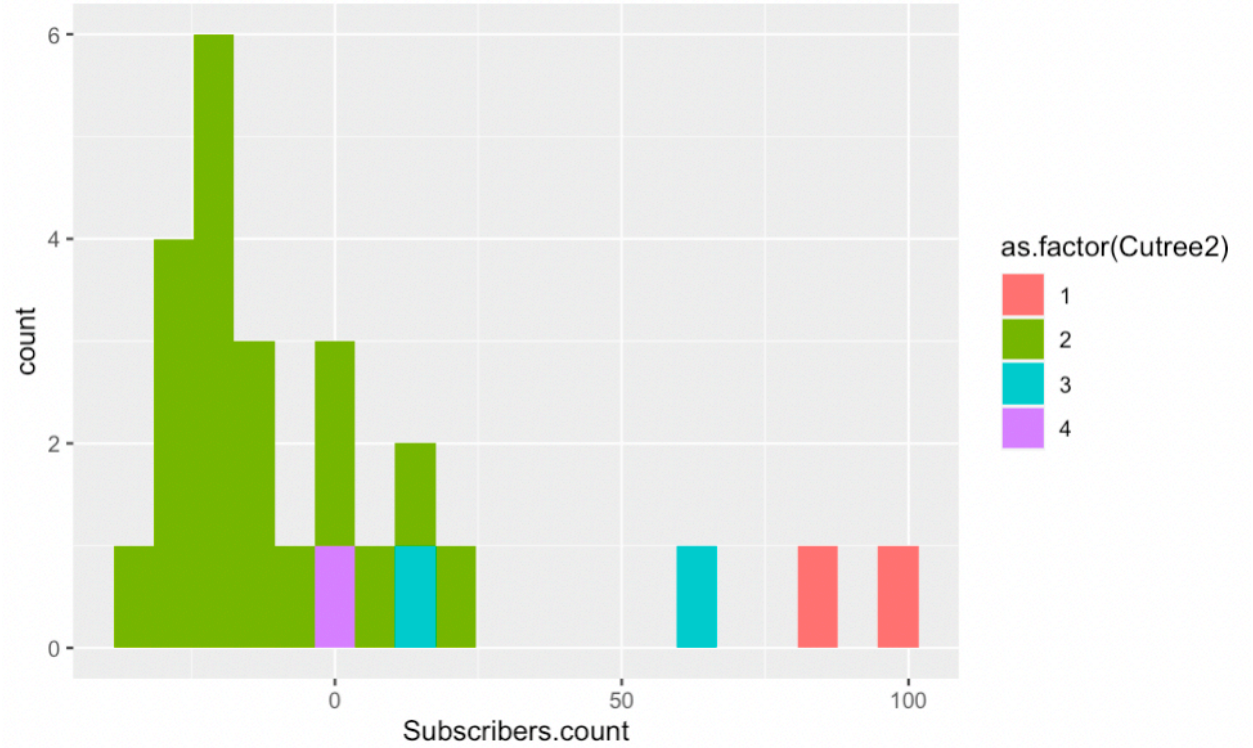
```
# A tibble: 4 × 7
  Cutree2 Count Sub_cnt View_avg Like_avg Comment_avg Share_avg
  <chr>   <int>   <dbl>    <dbl>    <dbl>       <dbl>     <dbl>
1 1           2   91.6     -10.7    -3.85       -4.10    -3.16
2 2          20  -13.3     -11.3    -3.19       -1.93     0.369
3 3           2   39.9     100.     28.1         5.05    -0.907
4 4           1    3.35     46.6    15.3        36.7      0.753
```

```r
# Cluster 1 has the highest subscribers count. Cluster 3 has the highest views
# average. Cluster 2 has the most artists but subscribers count, views average,
# likes average, and comments average are below average.
```

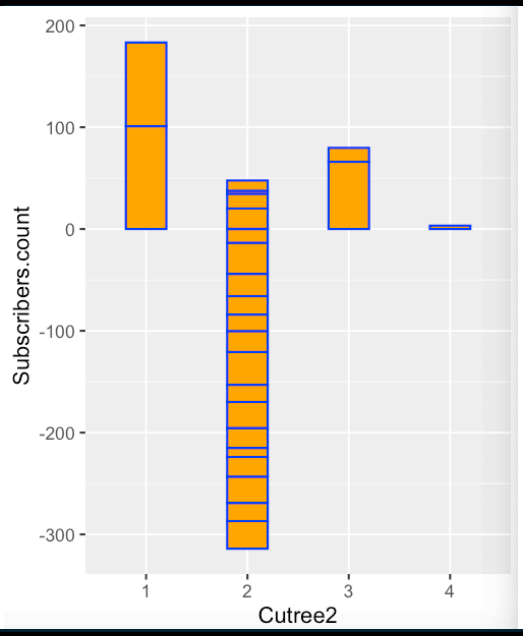
```r
# 8. d
# geom point
ggplot(ttw, aes(Cutree2, Subscribers.count)) +
  geom_point(color = "orange")
```




```r
# geom histogram
ggplot(ttw, aes(Subscribers.count, fill = as.factor(Cutree2))) +
  geom_histogram(bins = 20)
```



```r
# geom bar
ggplot(ttw, aes(Cutree2, Subscribers.count)) +
  geom_bar(width = 0.4, stat = "identity", color = "blue", fill = "orange")
```



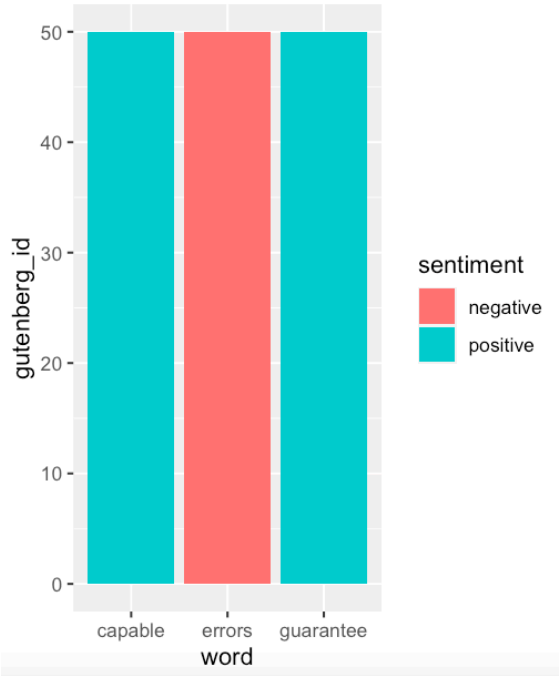

```r
# 8. e
# Fuji went from cluster 2 to cluster 3. Mackenzie Ziegler is in the same
# cluster with Fuji for both model. The order of the cluster changed, but the
# artists in it are the same. These two artists have the most high views
# average. The change of cluster order could cause by increasing the weight on
# views average.

# Task 2: Text Mining
# 1.
library(gutenbergr)
library(tidytext)
library(wordcloud)
library(textdata)
View(gutenberg_works())

# 2.
const <- gutenberg_download(50)

# 3.
View(const)
# I see 28,902 rows of data with gutenberg_id equal to 50.
```

```r
# 4. a
tidyconst <- const %>% unnest_tokens(word, text)
View(tidyconst)

# 4. b
# Rows increased from 28,902 to 125,749. Only lower case, and one word or number
# in a single square.
```

```
> # 5.
> tidyconst %>% count(word, sort = TRUE)
# A tibble: 125,633 × 2
   word        n
   <chr>   <int>
 1 the        16
 2 of         11
 3 a           8
 4 digits      8
 5 in          8
 6 and         6
 7 to          6
 8 from        5
 9 million     5
10 we          5
# … with 125,623 more rows
> # 5. a
> tidyconst %>%  anti_join(stop_words) %>% count(word, sort = TRUE)
Joining, by = "word"
# A tibble: 125,561 × 2
   word           n
   <chr>      <int>
 1 digits         8
 2 million        5
 3 hemphill       4
 4 1,000          3
 5 forwarded      3
 6 scott          3
 7 9350758837     2
 8 binary         2
 9 check          2
10 checked        2
# … with 125,551 more rows
```

```r
# 5. b
tidyconst2 <- const %>% unnest_tokens(bigram, text, token="ngrams", n=2)
# tidyconst2 <- drop_na(tidyconst2)
View(tidyconst2)

# 5. b. i
# Bigrams is two word in one single square, unigrams is one word in one single
# square.

# 5. b. ii
# The computer will learned the language differently. In unigram, the computer
# will consider word one by one, and in bigram it will consider two words at a
# time. In bigram the results are repeated, such as 'from scott' then 'scott
# hemphill'.

# 5. c
# I was preparing GRE test last year. I think the GRE vocabulary flashcards
# or GRE vocabulary books were created using unigrams.
```

```r
# 6. a
View(tidyconst)
sa <- tidyconst %>% inner_join(get_sentiments("bing"))
View(sa)

# 6. b
sa
> sa
# A tibble: 3 × 3
  gutenberg_id word      sentiment
         <int> <chr>     <chr>
1           50 capable   positive
2           50 guarantee positive
3           50 errors    negative
# There are only 3 words in my seed value of download. Two positive, one
# negative.

# 6. c
# The simple size is very small, but I will say this list could suggests to
# software engineers. It has "errors", "capable", and "guarantee".

# 7.
ggplot(sa, aes(word, gutenberg_id, fill = sentiment)) +
  geom_bar(stat = 'identity')
```




```r
# 7. a
# Since I only have three words with sentiment, I see three bars in the plot
# two positive sentiment, and one negative sentiment.
```

```r
# 8.
sa2 <- tidyconst %>% inner_join(get_sentiments("afinn"))
View(sa2)

# 8. a
sum(sa2$value)
# The total value is 4

# 8. b
# I only have 8 words in my book, I would say the sample size is still small.
# It suggests my book is neutral and a little towards positive. It is helpful
# because it at least increased my simple size. But it still too small.
```

```
> sa2
# A tibble: 8 × 3
  gutenberg_id word      value
         <int> <chr>     <dbl>
1           50 capable       1
2           50 no           -1
3           50 agree         1
4           50 fit           1
5           50 hope          2
6           50 guarantee     1
7           50 agreement     1
8           50 errors       -2
```