Project

2023-04-25

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
df <- readr::read_csv("Movie-Dataset-Latest.csv",show_col_types = FALSE)
```

```
## New names:
## • `` -> `...1`
```

```
head(df)
```

```
## # A tibble: 6 × 9
##    ...1     id title            release_…¹ overv…² popul…³ vote_…⁴ vote_…⁵ video
##   <dbl>  <dbl> <chr>            <date>     <chr>     <dbl>   <dbl>   <dbl> <lgl>
## 1     0  19404 Dilwale Dulhani… 1995-10-20 Raj is…    25.9     8.7    3304 FALSE
## 2     1    278 The Shawshank R… 1994-09-23 Framed…    60.1     8.7   20369 FALSE
## 3     2    238 The Godfather    1972-03-14 Spanni…    62.8     8.7   15219 FALSE
## 4     3 724089 Gabriel's Infer… 2020-07-31 Profes…    28.3     8.6    1360 FALSE
## 5     4    424 Schindler's List 1993-11-30 The tr…    38.7     8.6   12158 FALSE
## 6     5 696374 Gabriel's Infer… 2020-05-29 An int…    18.4     8.6    2172 FALSE
## # … with abbreviated variable names ¹release_date, ²overview, ³popularity,
## #   ⁴vote_average, ⁵vote_count
```

```
#Summary of dataset
str(df)
```

```
## spc_tbl_ [9,463 × 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1        : num [1:9463] 0 1 2 3 4 5 6 7 8 9 ...
##  $ id          : num [1:9463] 19404 278 238 724089 424 ...
##  $ title       : chr [1:9463] "Dilwale Dulhania Le Jayenge" "The Shawshank Redemptio
n" "The Godfather" "Gabriel's Inferno Part II" ...
##  $ release_date: Date[1:9463], format: "1995-10-20" "1994-09-23" ...
##  $ overview    : chr [1:9463] "Raj is a rich, carefree, happy-go-lucky second generat
ion NRI. Simran is the daughter of Chaudhary Baldev Singh"| __truncated__ "Framed in the
1940s for the double murder of his wife and her lover, upstanding banker Andy Dufresne b
egins a n"| __truncated__ "Spanning the years 1945 to 1955, a chronicle of the fictional
Italian-American Corleone crime family. When orga"| __truncated__ "Professor Gabriel Eme
rson finally learns the truth about Julia Mitchell's identity, but his realization comes
a"| __truncated__ ...
##  $ popularity  : num [1:9463] 25.9 60.1 62.8 28.3 38.7 ...
##  $ vote_average: num [1:9463] 8.7 8.7 8.7 8.6 8.6 8.6 8.6 8.6 8.6 8.6 ...
##  $ vote_count  : num [1:9463] 3304 20369 15219 1360 12158 ...
##  $ video       : logi [1:9463] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   id = col_double(),
##   ..   title = col_character(),
##   ..   release_date = col_date(format = ""),
##   ..   overview = col_character(),
##   ..   popularity = col_double(),
##   ..   vote_average = col_double(),
##   ..   vote_count = col_double(),
##   ..   video = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(df)
```

```
##       ...1            id            title            release_date
##  Min.   :   0   Min.   :     5   Length:9463        Min.   :1902-04-17
##  1st Qu.:2366   1st Qu.:  9951   Class :character   1st Qu.:1997-04-07
##  Median :4731   Median : 25846   Mode  :character   Median :2008-10-22
##  Mean   :4731   Mean   :149335                      Mean   :2003-09-12
##  3rd Qu.:7096   3rd Qu.:290548                      3rd Qu.:2015-08-28
##  Max.   :9462   Max.   :876716                      Max.   :2021-12-16
##    overview           popularity         vote_average     vote_count
##  Length:9463        Min.   :    0.600   Min.   :2.200   Min.   :  200
##  Class :character   1st Qu.:    8.835   1st Qu.:6.100   1st Qu.:  316
##  Mode  :character   Median :   12.636   Median :6.600   Median :  584
##                     Mean   :   35.678   Mean   :6.597   Mean   : 1515
##                     3rd Qu.:   24.371   3rd Qu.:7.200   3rd Qu.: 1434
##                     Max.   :14136.690   Max.   :8.700   Max.   :30535
##    video
##  Mode :logical
##  FALSE:9463
##
##
##
##
```

```
#Checking for null values
colSums(is.na(df))
```

```
##         ...1           id        title release_date     overview   popularity
##            0            0            0            0           14            0
## vote_average   vote_count        video
##            0            0            0
```

```
#Converting the column type of release_date to date type-Already the present type is dat
e type-no need of conversion
df$release_date <- as.Date(df$release_date, format = "%Y-%m-%d")
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
df$year <- year(as.Date(df$release_date))
df$month <- month(as.Date(df$release_date))
```

```
df1 <- df[, !(names(df) %in% c('Unnamed: 0', 'id', 'release_date', 'overview', 'video',
'year', 'month','title'))]
```

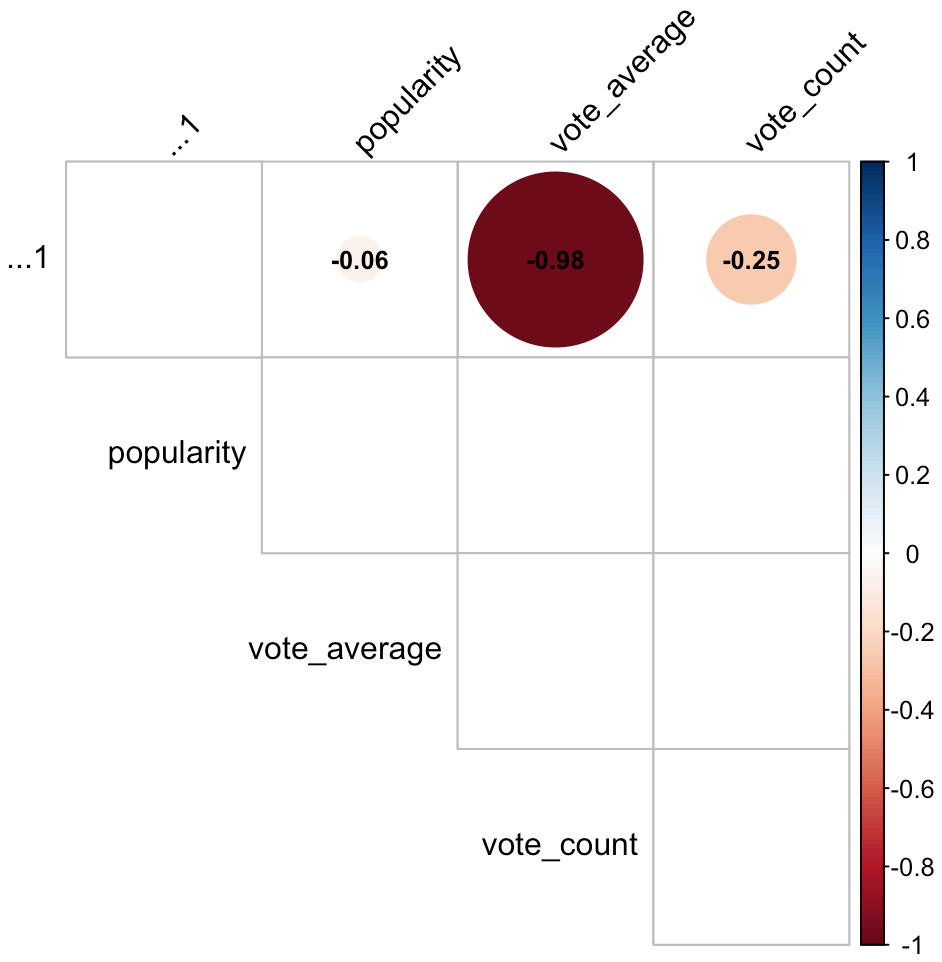
```
df_100 <- head(df[order(-df$popularity), ], 100)
```

#DATA VISUALIZATION

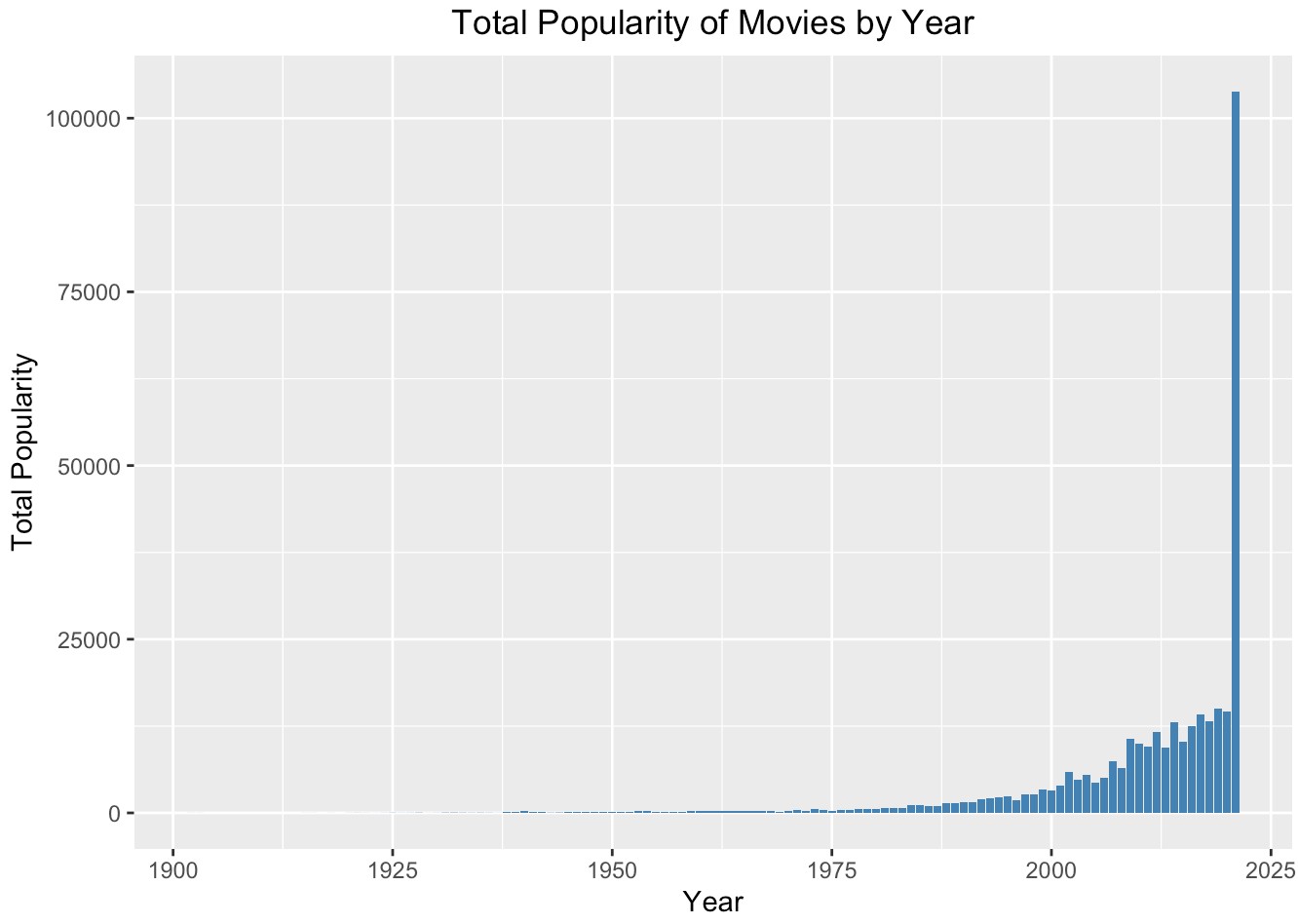
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
# Compute correlation matrix
corr_matrix <- cor(df1)

# Create correlation plot with annotations
corrplot(corr_matrix, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45,
         method = "circle", number.cex = 0.8,
         addCoef.col = "black",
         p.mat = corr_matrix, sig.level = 0.05, insig = "blank")
```



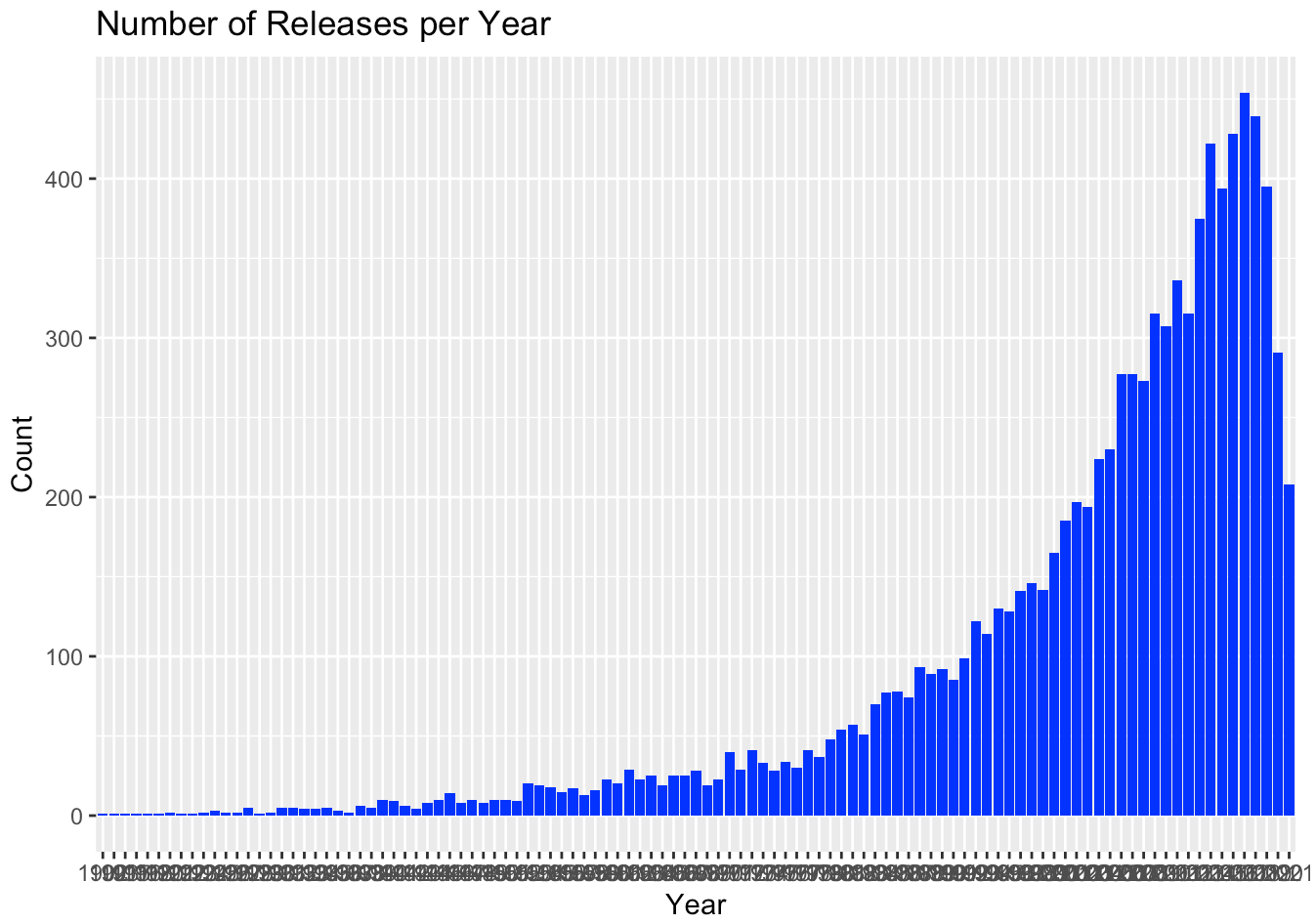
```
library(ggplot2)

# Create data frame with total popularity by year
df_year <- aggregate(df$popularity, by = list(df$year), FUN = sum)
names(df_year) <- c("year", "total_popularity")

# Create bar plot
ggplot(df_year, aes(x = year, y = total_popularity)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  ggtitle("Total Popularity of Movies by Year") +
  xlab("Year") +
  ylab("Total Popularity") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
head(df_100, 10)
```

```
## # A tibble: 10 × 11
##     ...1     id title     release_…¹ overv…² popul…³ vote_…⁴ vote_…⁵ video  year
##    <dbl>  <dbl> <chr>     <date>     <chr>     <dbl>   <dbl>   <dbl> <lgl> <dbl>
##  1    23 634649 Spider-M… 2021-12-15 Peter …  14137.     8.5    2654 FALSE  2021
##  2   691 568124 Encanto   2021-11-24 The ta…   8663.     7.8    1065 FALSE  2021
##  3  1748 624860 The Matr… 2021-12-16 Plague…   7976.     7.4    1029 FALSE  2021
##  4  2371 580489 Venom: L… 2021-09-30 After …   7537.     7.2    5184 FALSE  2021
##  5  6333 460458 Resident… 2021-11-24 Once t…   4974.     6.2     456 FALSE  2021
##  6  3796 512195 Red Noti… 2021-11-04 An Int…   3645.     6.8    2294 FALSE  2021
##  7   587 566525 Shang-Ch… 2021-09-01 Shang-…   2968.     7.8    4608 FALSE  2021
##  8  4674   1930 The Amaz… 2012-06-23 Peter …   2514.     6.6   13992 FALSE  2012
##  9  1715 315635 Spider-M… 2017-07-05 Follow…   2475.     7.4   17559 FALSE  2017
## 10  1581 585245 Clifford… 2021-11-10 As Emi…   2312.     7.4     712 FALSE  2021
## # … with 1 more variable: month <dbl>, and abbreviated variable names
## #   ¹release_date, ²overview, ³popularity, ⁴vote_average, ⁵vote_count
```

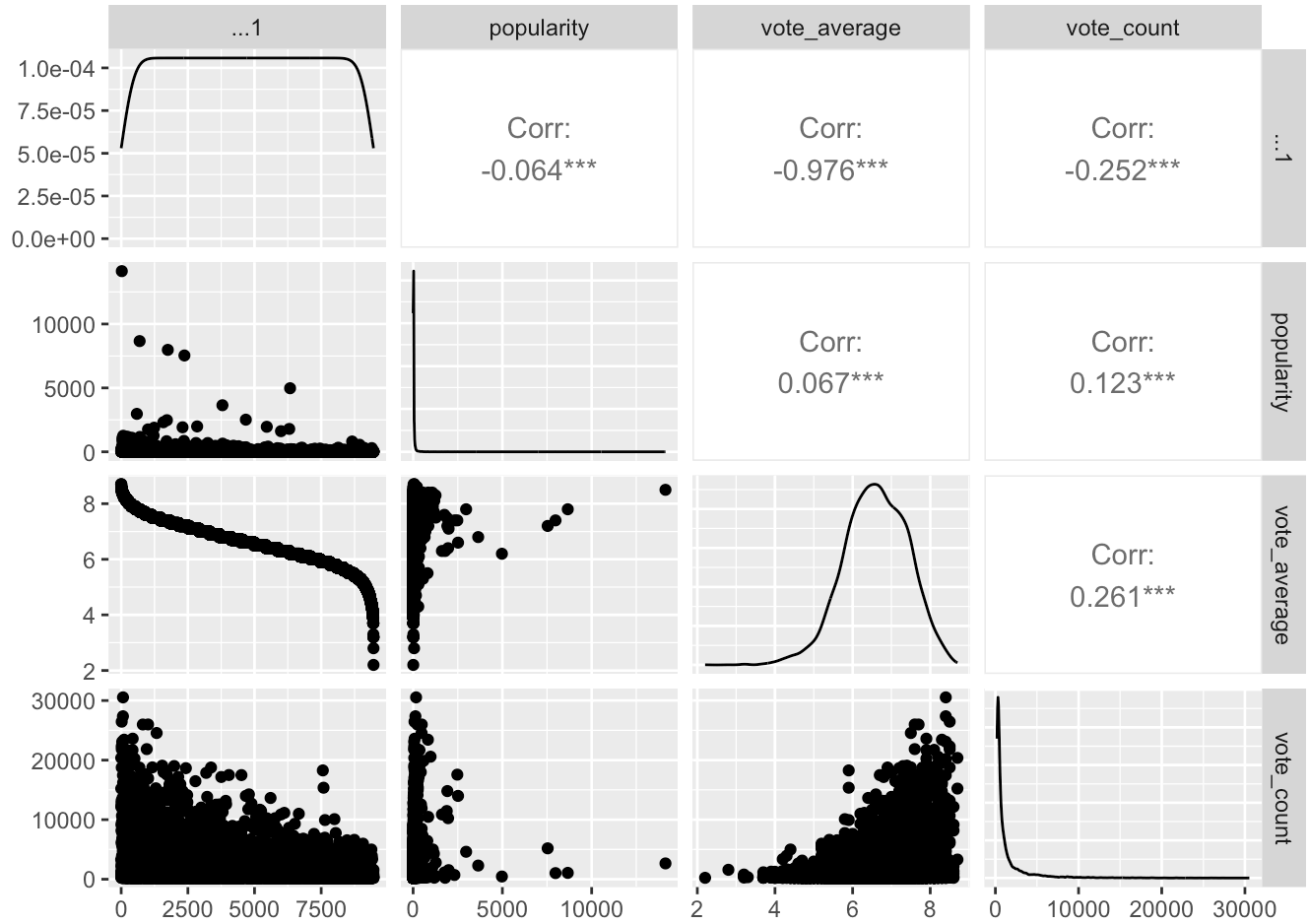
```
library(ggplot2)

# Create a data frame with the count of releases per year
df_year <- data.frame(table(df$year))

# Plot a histogram with ggplot2
ggplot(df_year, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "blue") +
  ggtitle("Number of Releases per Year") +
  xlab("Year") + ylab("Count")
```




```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(data = df1)
```

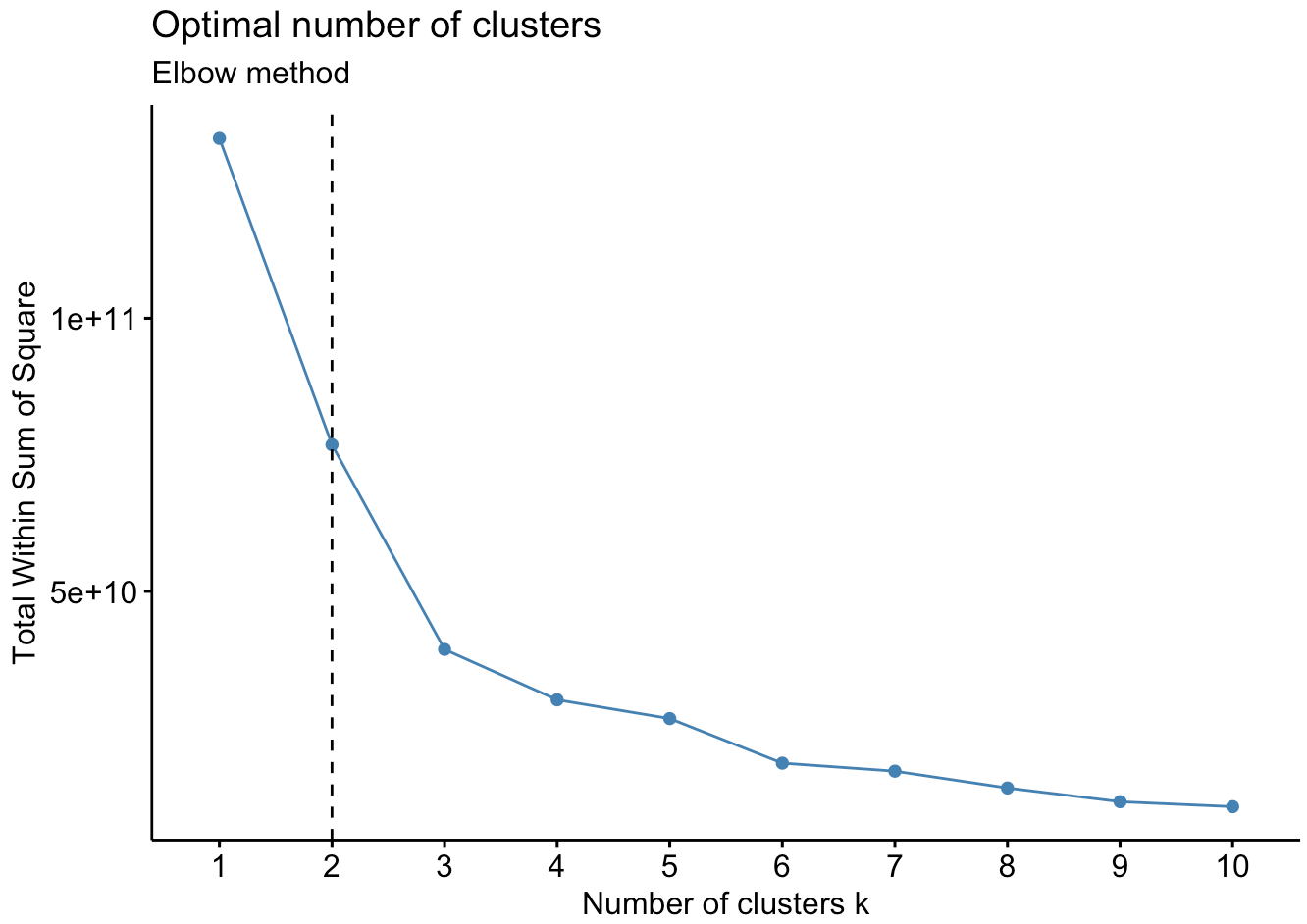


#kmeans clustering

```
library(factoextra)
```

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

```
set.seed(1)
fviz_nbclust(df1, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```
# Set the number of clusters
num_clusters <- 3

# Run k-means clustering
kmeans_model <- kmeans(df1, centers = num_clusters)

# Print the summary of the model
summary(kmeans_model)
```

```
##              Length Class  Mode
## cluster      9463   -none- numeric
## centers        12   -none- numeric
## totss           1   -none- numeric
## withinss        3   -none- numeric
## tot.withinss    1   -none- numeric
## betweenss       1   -none- numeric
## size            3   -none- numeric
## iter            1   -none- numeric
## ifault          1   -none- numeric
```

```
#any(is.na(df))
#apply(df, 2, function(x) any(is.infinite(x)))
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.3.0      ✔ stringr 1.5.0
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ✔ purrr   1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ lubridate::as.difftime() masks base::as.difftime()
## ✖ lubridate::date()        masks base::date()
## ✖ dplyr::filter()          masks stats::filter()
## ✖ lubridate::intersect()   masks base::intersect()
## ✖ dplyr::lag()             masks stats::lag()
## ✖ lubridate::setdiff()     masks base::setdiff()
## ✖ lubridate::union()       masks base::union()
```

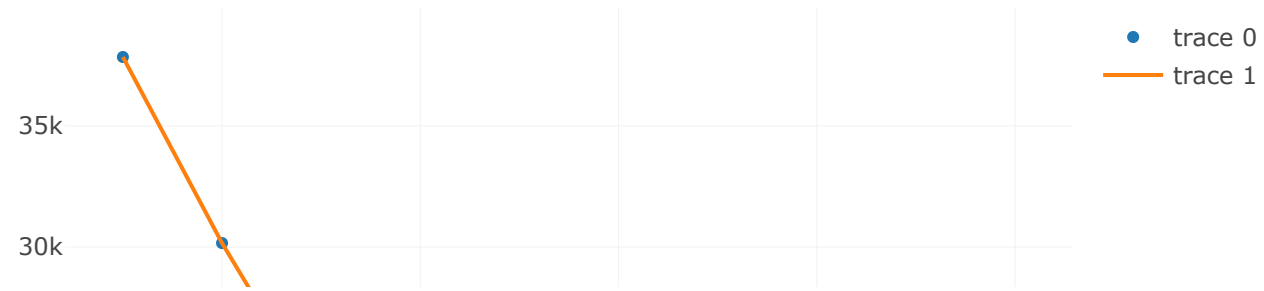
```
library(cluster)
#library(factoextra)
library(lubridate)
library(plotly)
```

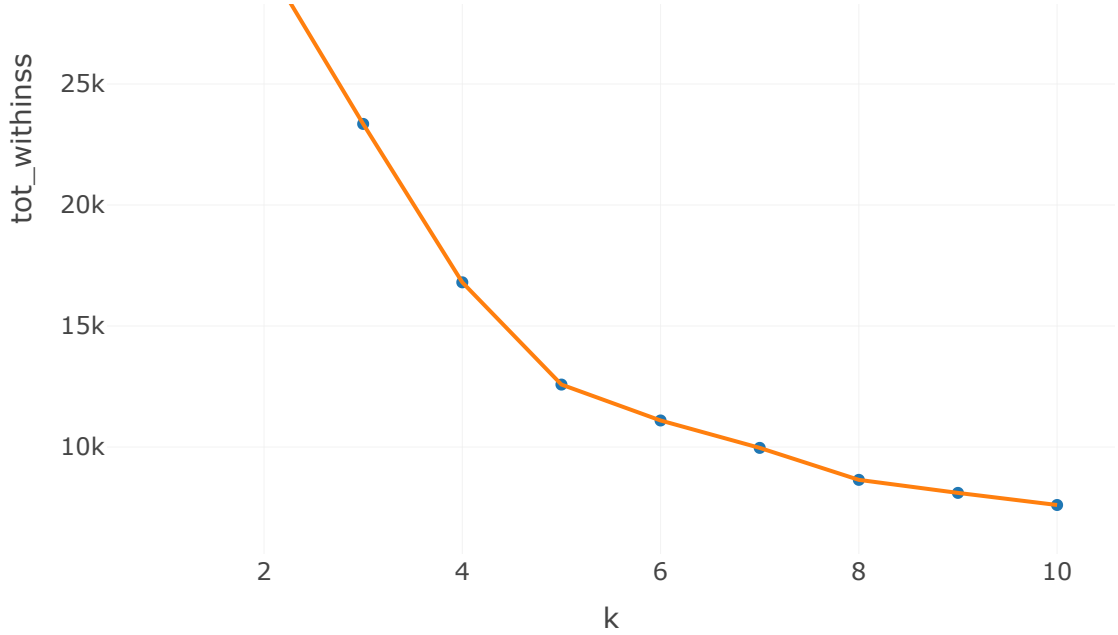
```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
movies <- df
# Extracting year from released date
movies$year <- year(movies$release_date)

# Selecting variables
movie_data <- movies %>%
  select(year,popularity, vote_average, vote_count) %>%
  drop_na()

# Standardizing the variables
movie_data_scaled <- scale(movie_data)

# optimal number of clusters using the elbow method
#elbow_plot <- fviz_nbclust(movie_data_scaled, kmeans, method = "wss") +
#  labs(title = "Elbow plot")


km_out_list <- lapply(1:10, function(k) list(
  k=k,
  km_out=kmeans(movie_data_scaled, k, nstart = 20)))

km_results <- data.frame(
  k=sapply(km_out_list, function(k) k$k),
  totss=sapply(km_out_list, function(k) k$km_out$totss),
  tot_withinss=sapply(km_out_list, function(k) k$km_out$tot.withinss)
  )
km_results
```

```
##     k totss tot_withinss
## 1   1 37848    37848.000
## 2   2 37848    30159.749
## 3   3 37848    23349.997
## 4   4 37848    16804.843
## 5   5 37848    12580.500
## 6   6 37848    11096.790
## 7   7 37848     9965.435
## 8   8 37848     8643.829
## 9   9 37848     8104.950
## 10 10 37848     7605.216
```

```
plot_ly(km_results,x=~k,y=~tot_withinss) %>% add_markers() %>% add_paths()
```

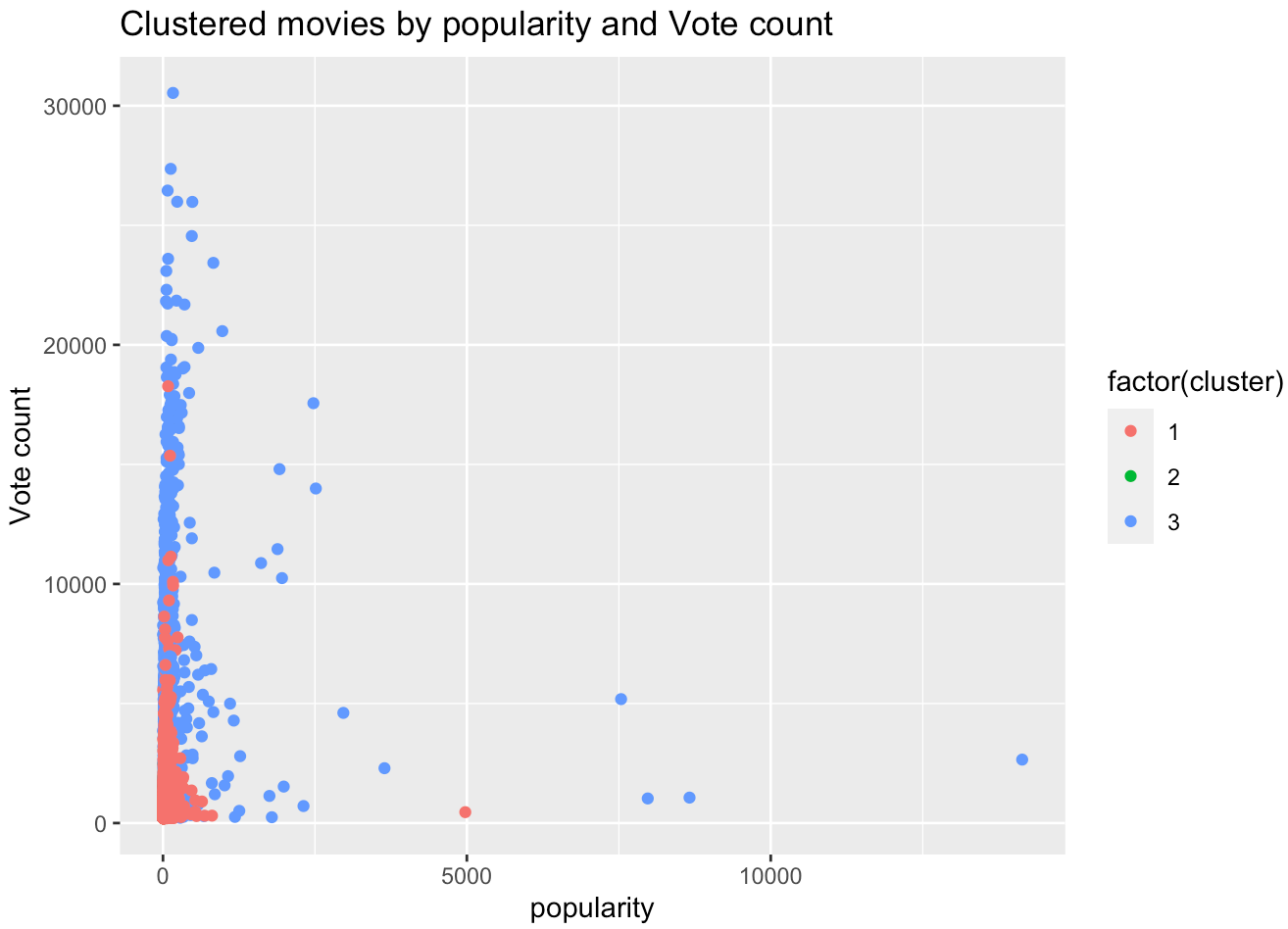




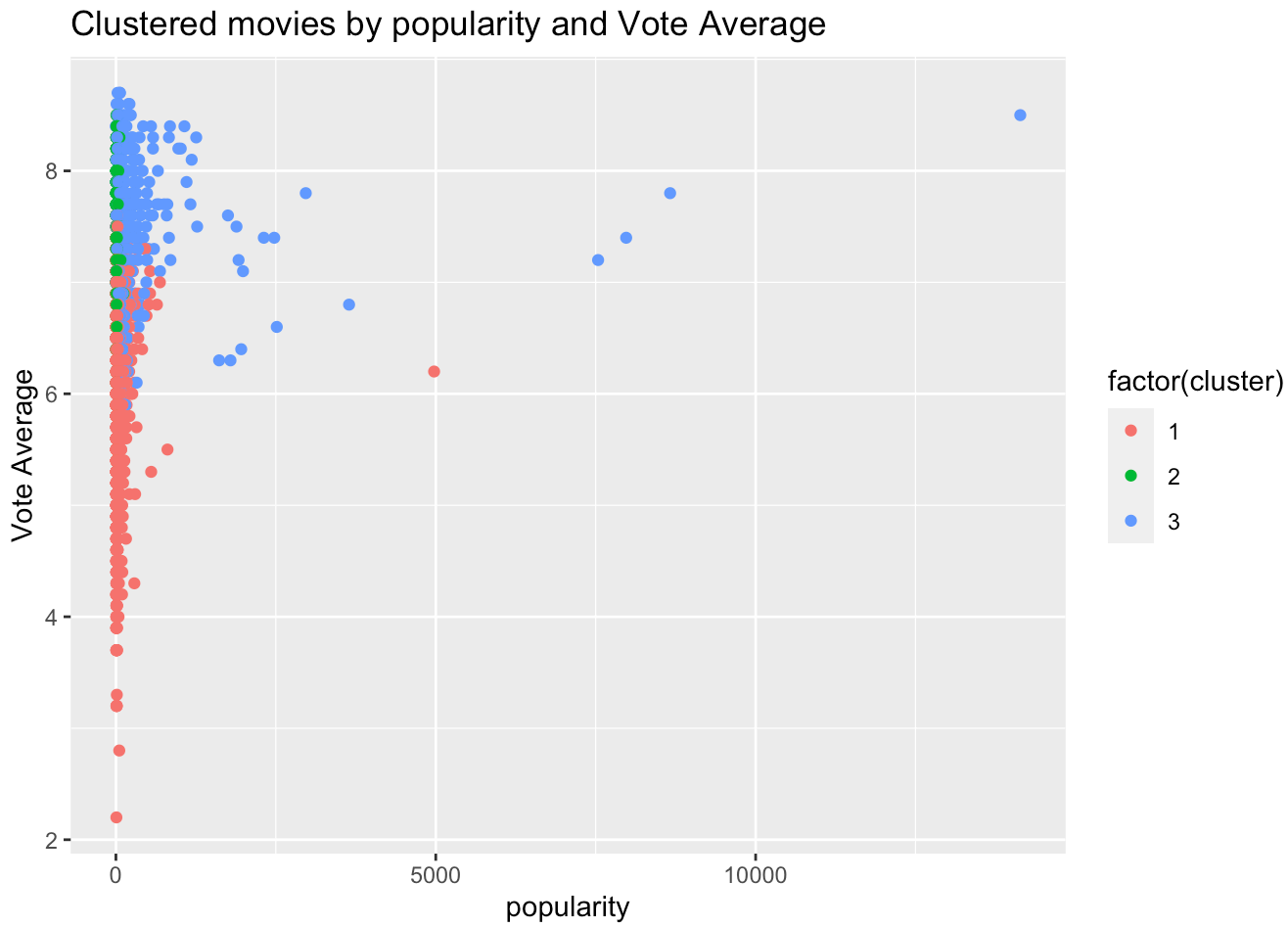

```
# Performing k-means clustering
k <- 3  #Taking number of clusters From elbow plot
kmeans_model <- kmeans(movie_data_scaled, centers = k)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 473150)
```

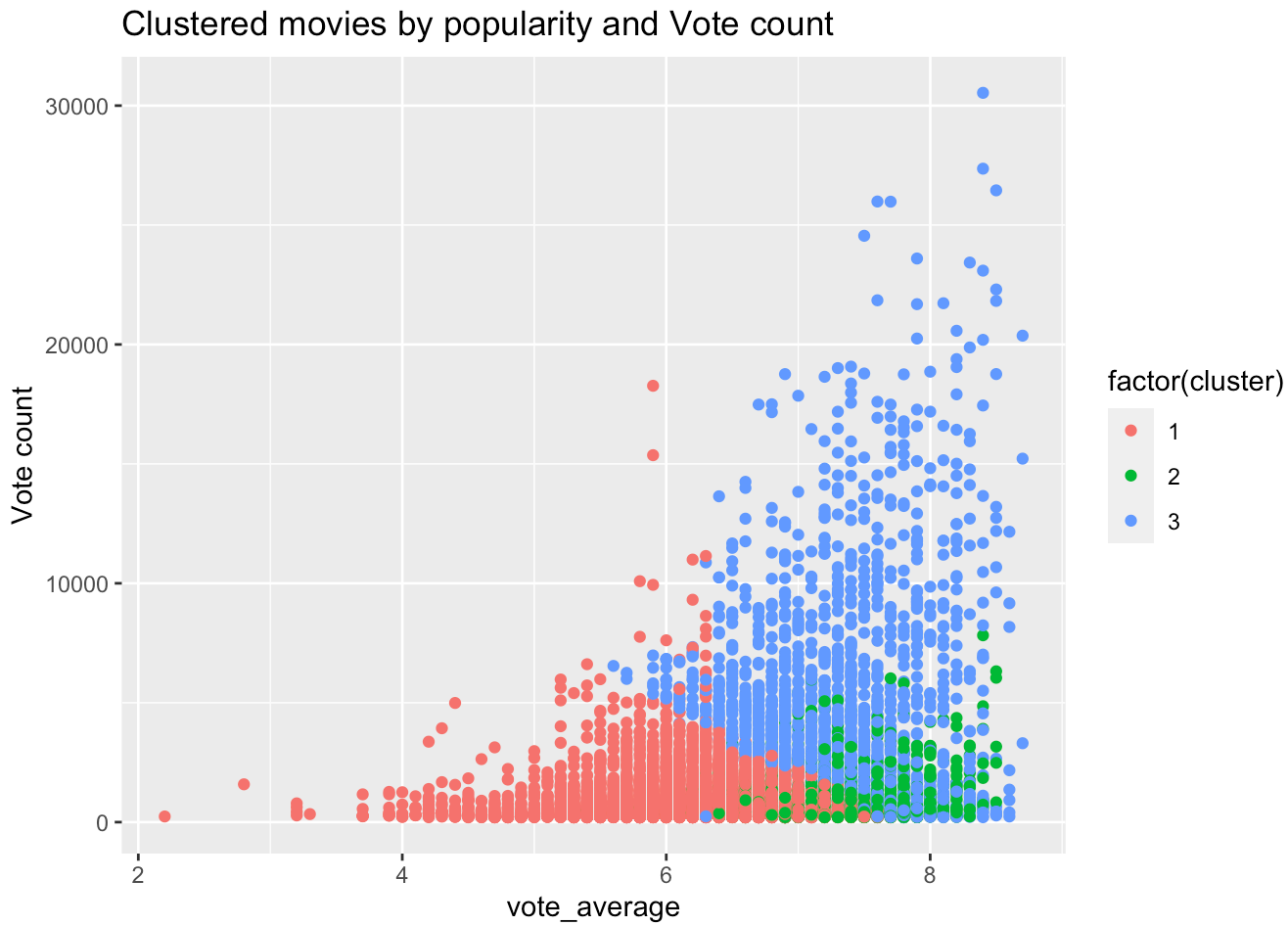
```
# Add cluster labels to the original data
movies_clustered <- movies %>%
  select(year,popularity, vote_average, vote_count) %>%
  drop_na() %>%
  mutate(cluster = kmeans_model$cluster)

ggplot(movies_clustered, aes(x = popularity, y = vote_count, color = factor(cluster))) +
  geom_point() +
  labs(x = "popularity", y = "Vote count") +
  ggtitle("Clustered movies by popularity and Vote count")
```



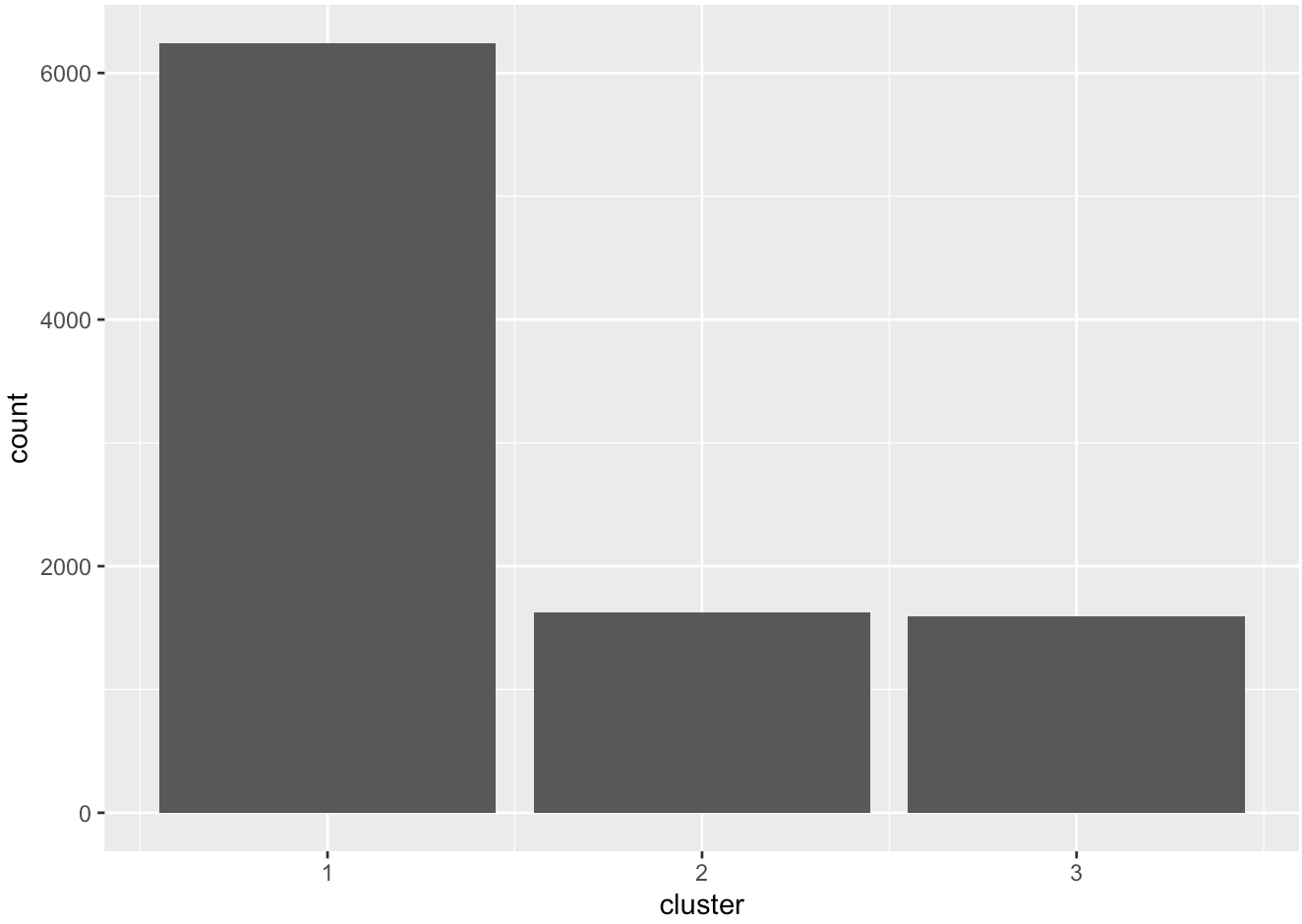

```
ggplot(movies_clustered, aes(x = popularity, y = vote_average, color = factor(cluster)))
+
  geom_point() +
  labs(x = "popularity", y = "Vote Average") +
  ggtitle("Clustered movies by popularity and Vote Average")
```



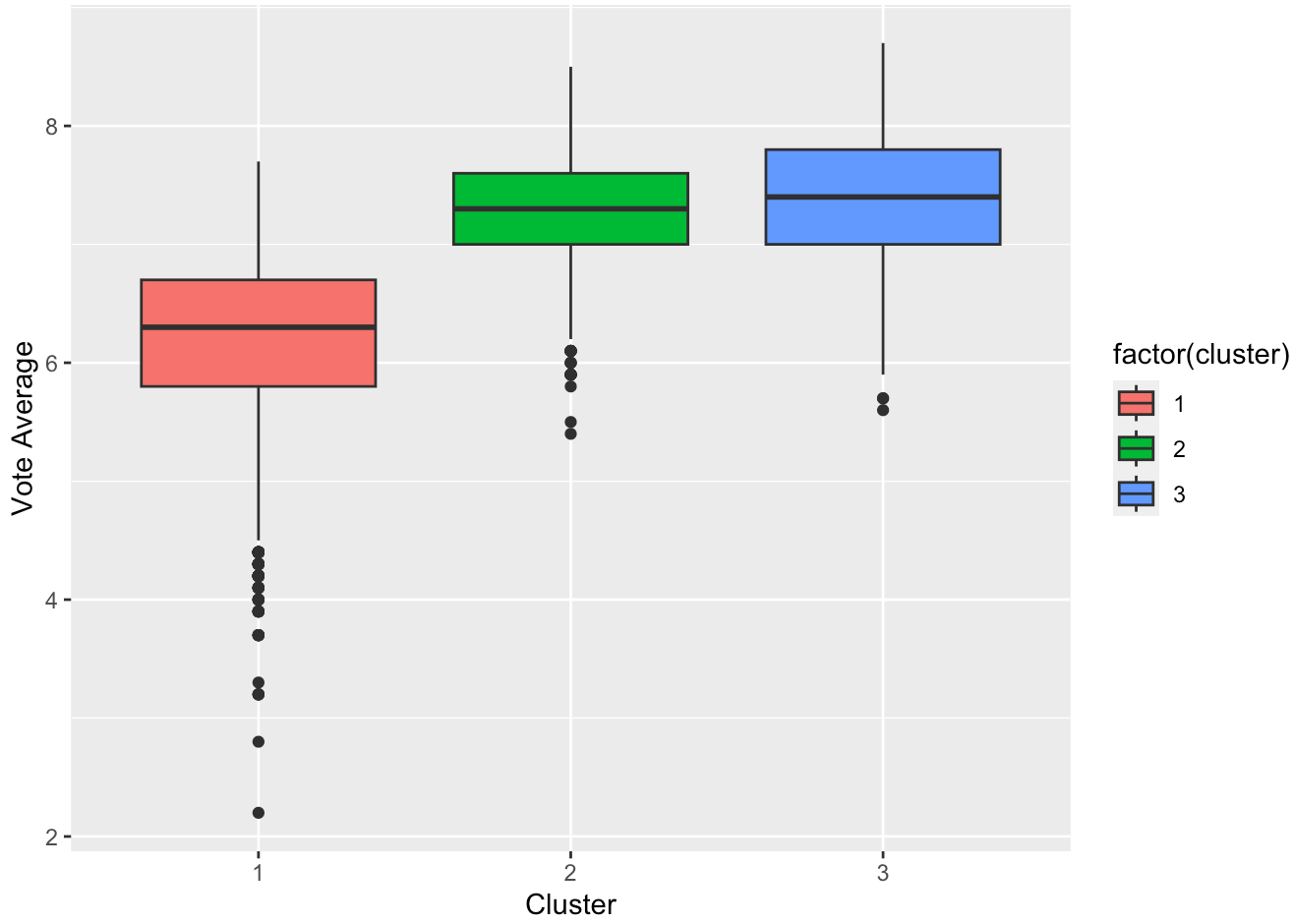

```
ggplot(movies_clustered, aes(x = vote_average, y = vote_count, color = factor(cluster)))
+
  geom_point() +
  labs(x = "vote_average", y = "Vote count") +
  ggtitle("Clustered movies by popularity and Vote count")
```



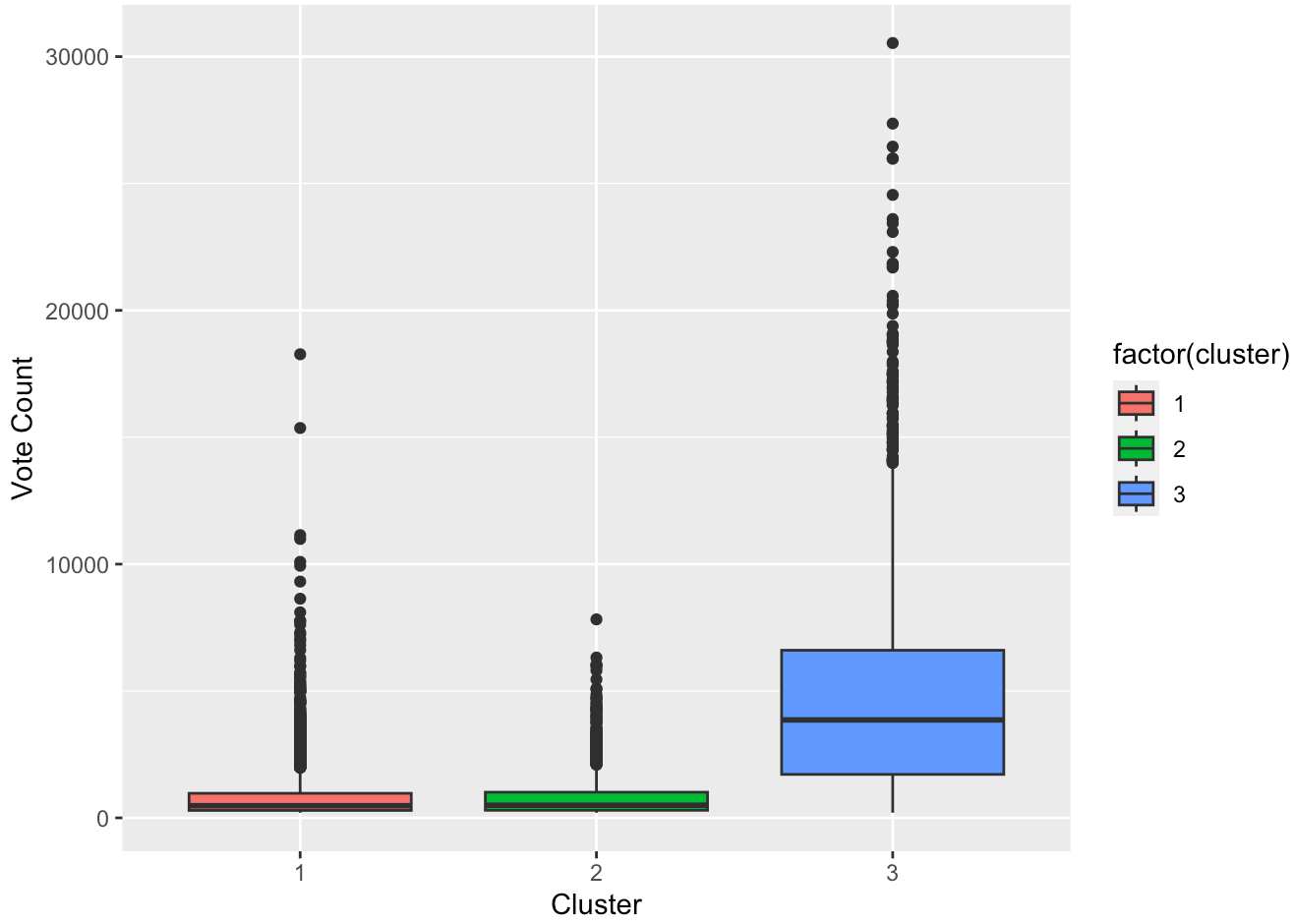

```
group_counts <- movies_clustered %>%
  group_by(cluster) %>%
  summarise(count = n())

# plot bar plot
ggplot(group_counts, aes(x = cluster, y = count)) +
  geom_bar(stat = "identity")
```



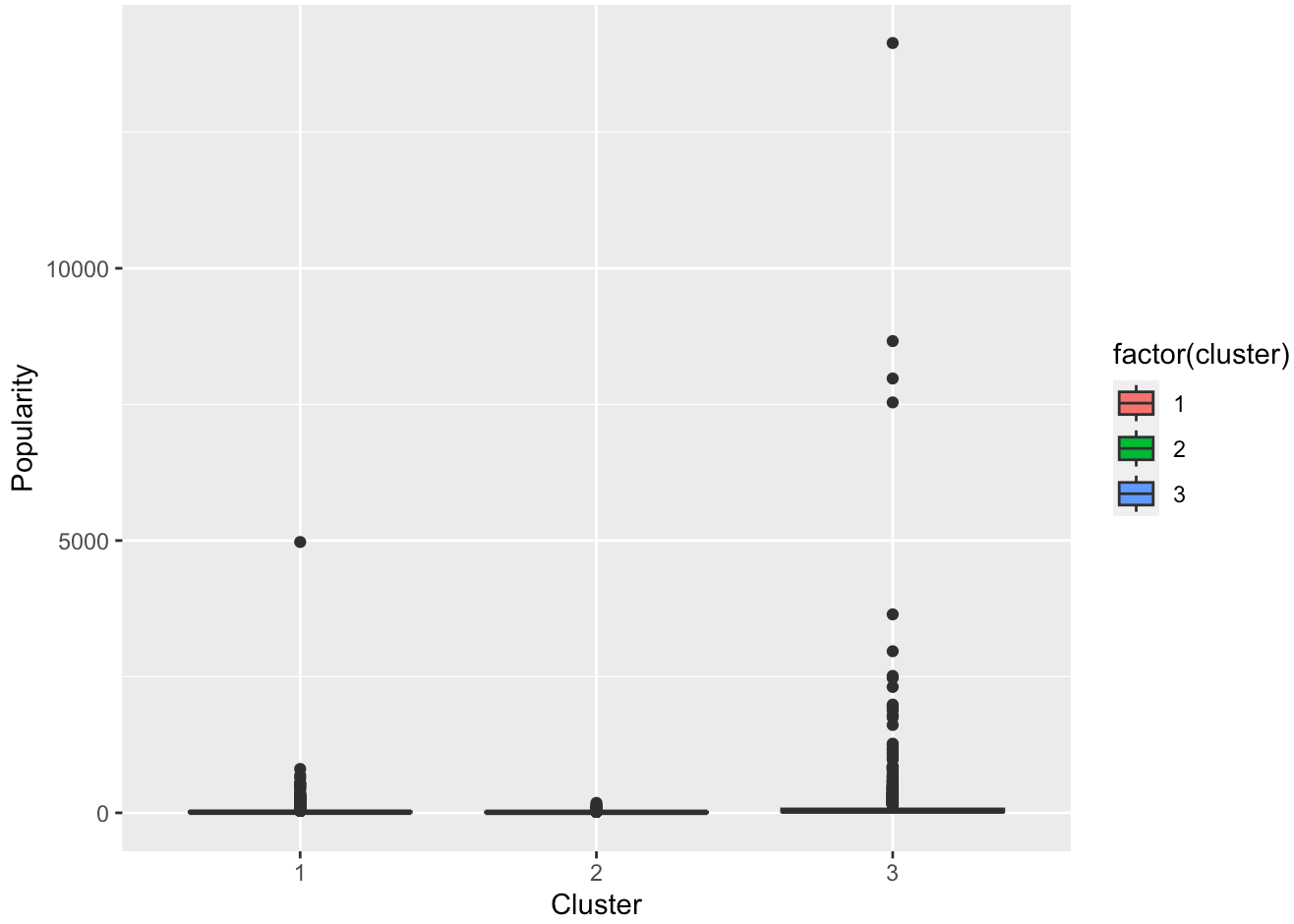

```
ggplot(movies_clustered, aes(x = factor(cluster), y = vote_average, fill = factor(cluste
r))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Vote Average")
```




```
ggplot(movies_clustered, aes(x = factor(cluster), y = vote_count, fill = factor(cluste
r))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Vote Count")
```

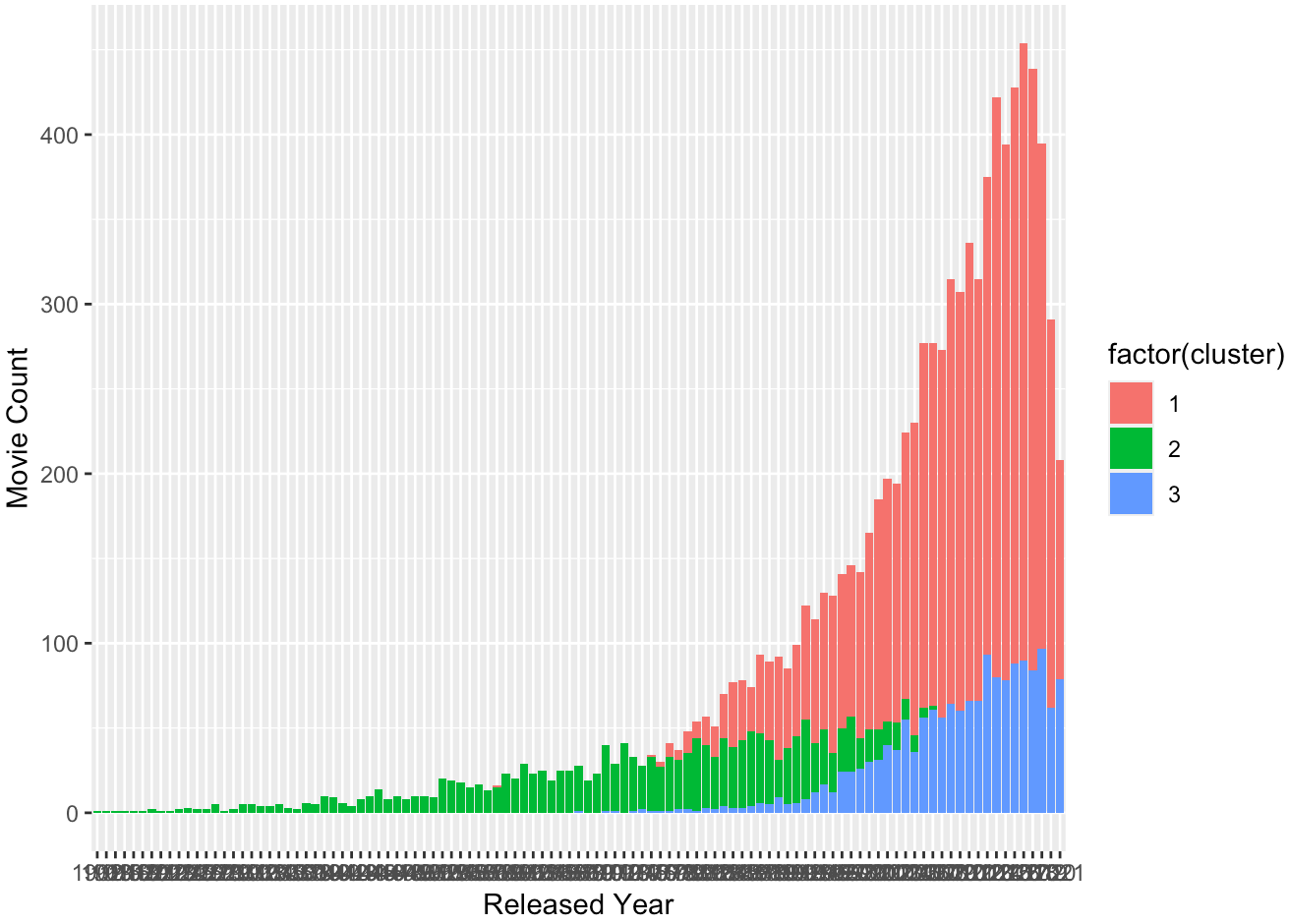



```
ggplot(movies_clustered, aes(x = factor(cluster), y = popularity, fill = factor(cluste
r))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Popularity")
```




```
movies_clustered %>%
  group_by(cluster, year) %>%
  summarise(count = n()) %>%
  ggplot(aes(x = factor(year), y = count, fill = factor(cluster))) +
  geom_bar(stat = "identity") +
  labs(x = "Released Year", y = "Movie Count")
```

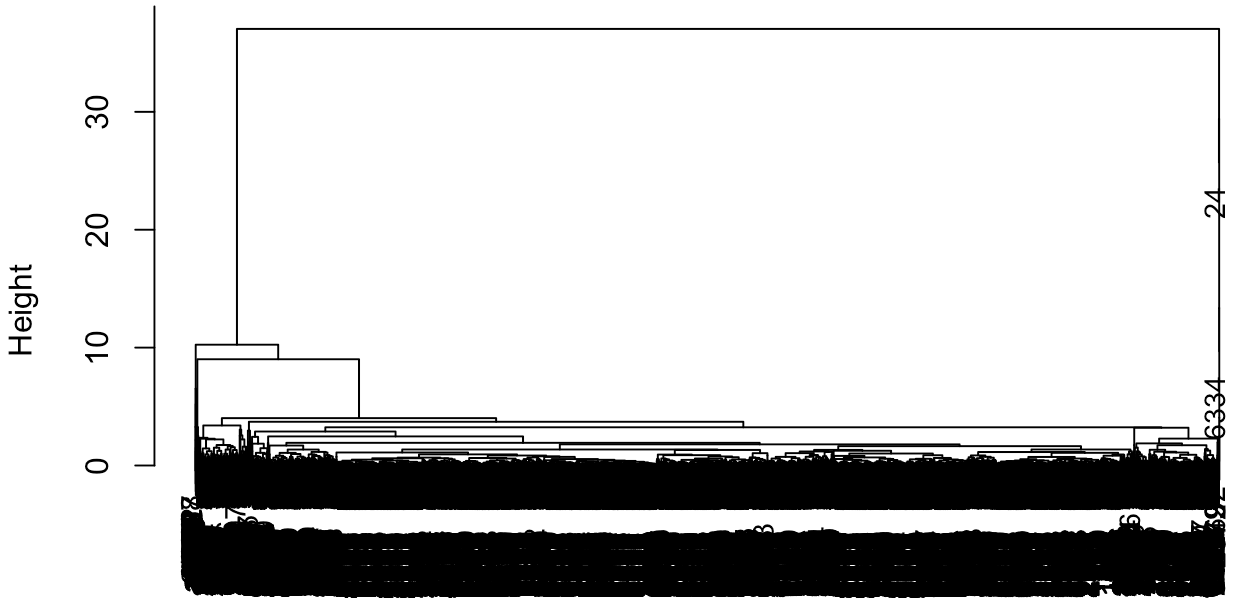
```
## `summarise()` has grouped output by 'cluster'. You can override using the
## `.groups` argument.
```




```
library(cluster)
# Compute the between-cluster sum of squares
between_cluster_sumsq <- sum(kmeans_model$betweenss)

# Print the between-cluster sum of squares
cat("Between-cluster sum of squares:", between_cluster_sumsq, "\n")
```

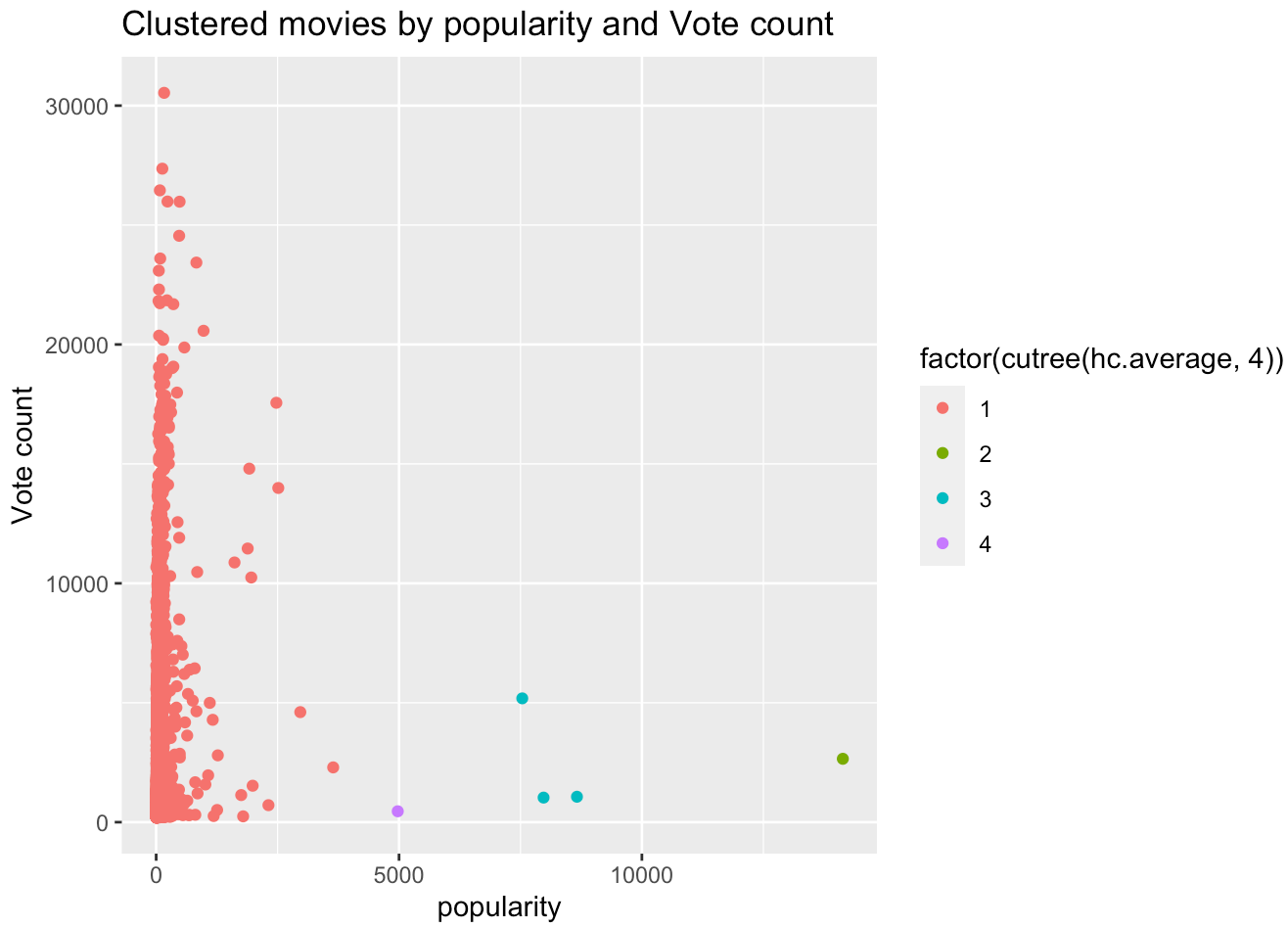
```
## Between-cluster sum of squares: 13050.57
```

```
set.seed(80)
x_dist<- dist(movie_data_scaled)
hc.average <- hclust(dist(movie_data_scaled), method = "average")
plot(hc.average, main = "Average Linkage", xlab = "", sub = "", cex = .9)
```

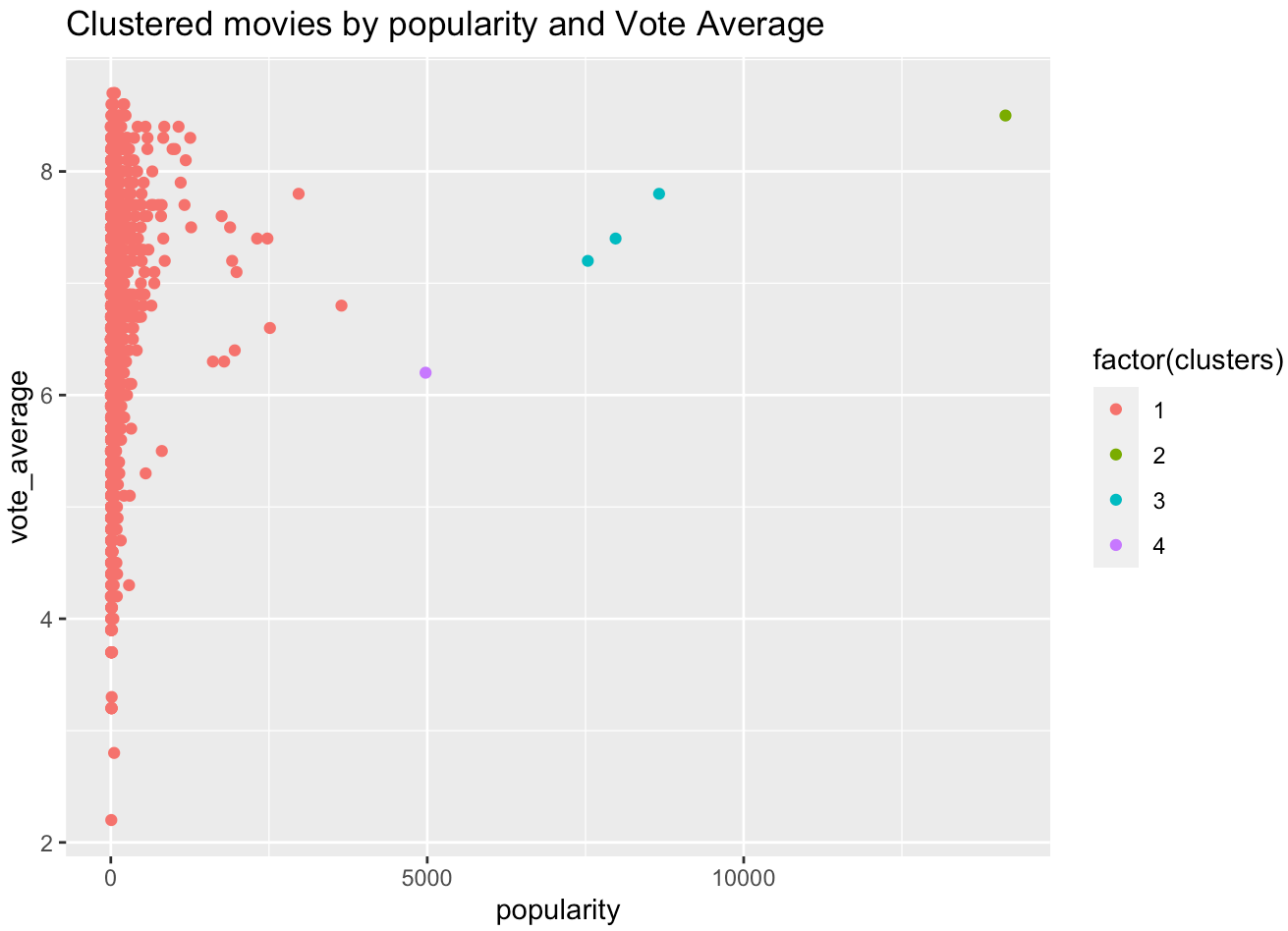



```
clusters <- cutree(hc.average,4)
```

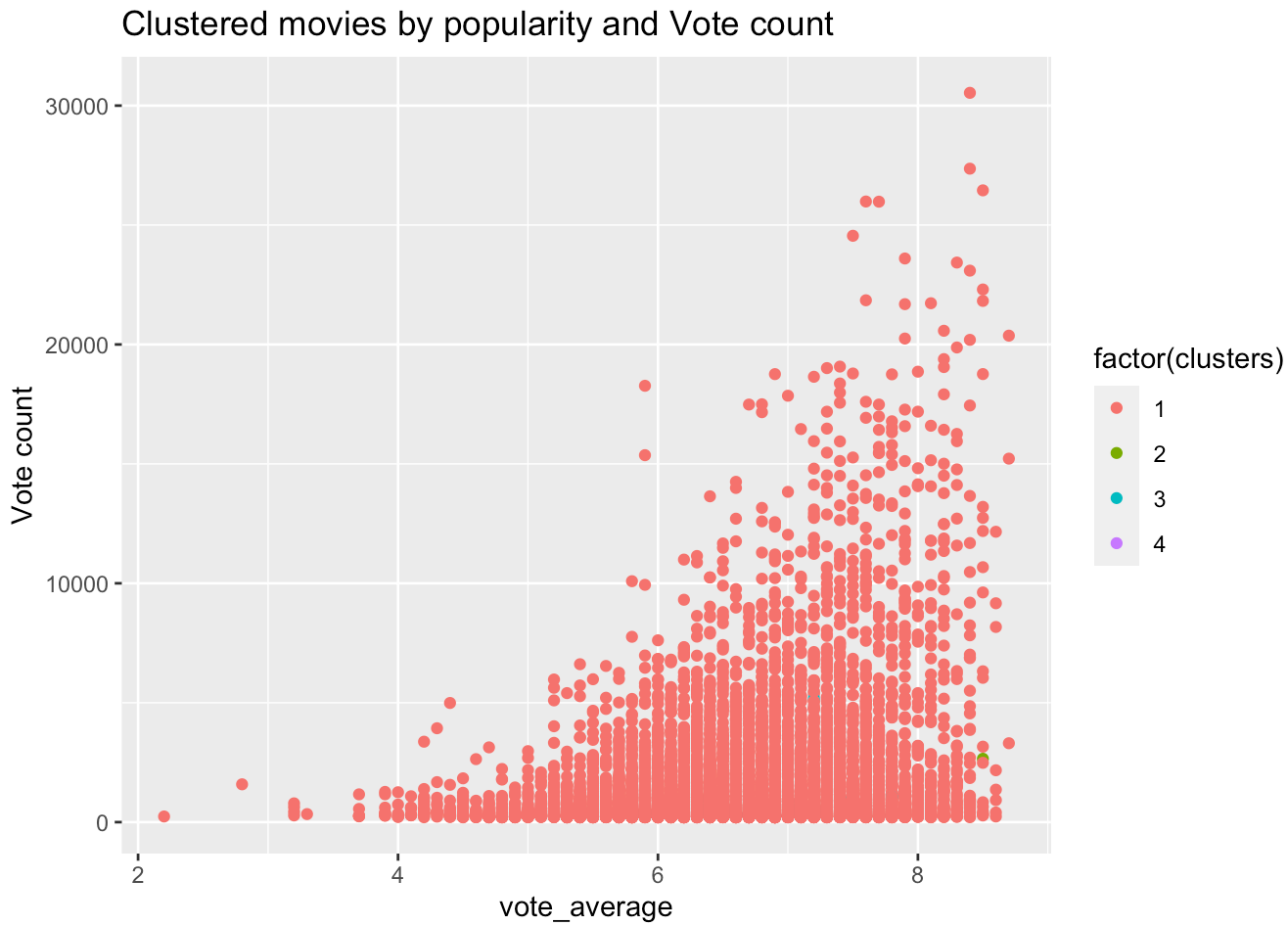
```
ggplot(movies, aes(x = popularity, y = vote_count, color = factor(cutree(hc.average,
4)))) +
  geom_point() +
  labs(x = "popularity", y = "Vote count") +
  ggtitle("Clustered movies by popularity and Vote count")
```




```
ggplot(movies, aes(x = popularity, y = vote_average, color = factor(clusters))) +
  geom_point() +
  labs(x = "popularity", y = "vote_average") +
  ggtitle("Clustered movies by popularity and Vote Average")
```




```
ggplot(movies_clustered, aes(x = vote_average, y = vote_count, color = factor(cluster
s))) +
  geom_point() +
  labs(x = "vote_average", y = "Vote count") +
  ggtitle("Clustered movies by popularity and Vote count")
```



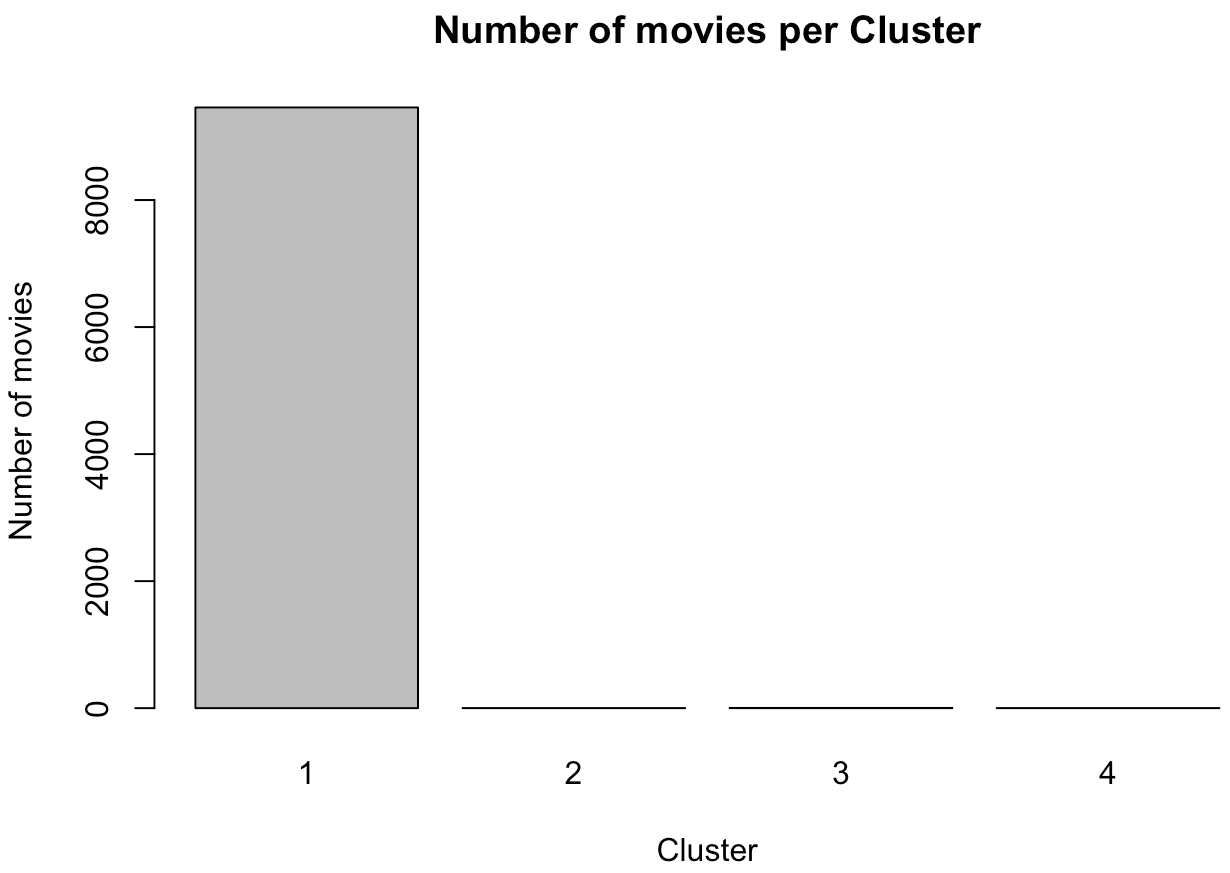

```
rows_per_cluster <- table(clusters)
cat("Number of rows per cluster:\n")
```

```
## Number of rows per cluster:
```

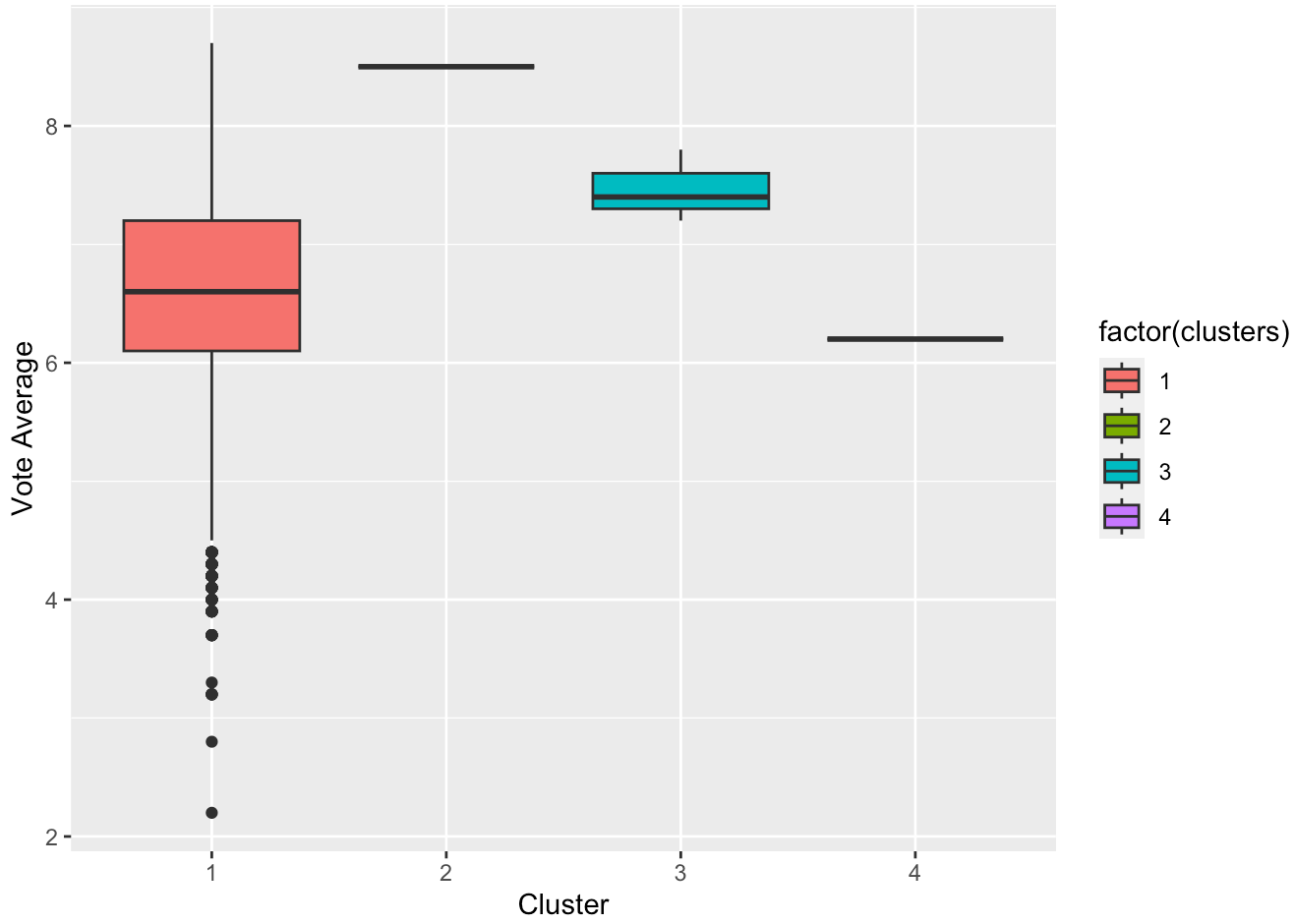
```
print(rows_per_cluster)
```

```
## clusters
##    1    2    3    4
## 9458    1    3    1
```

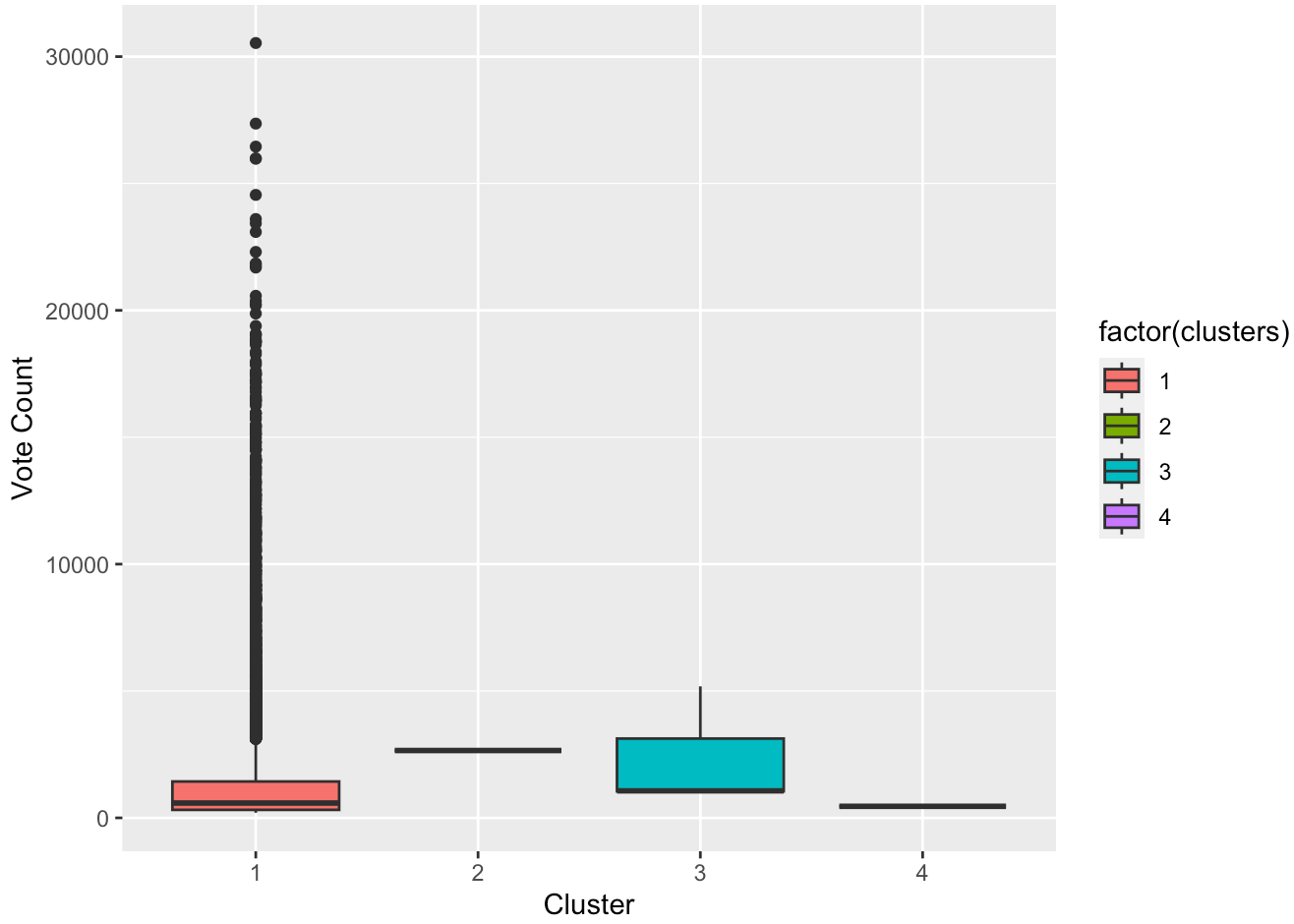
```
# plot bar plot
barplot(rows_per_cluster, main = "Number of movies per Cluster",
        xlab = "Cluster", ylab = "Number of movies")
```



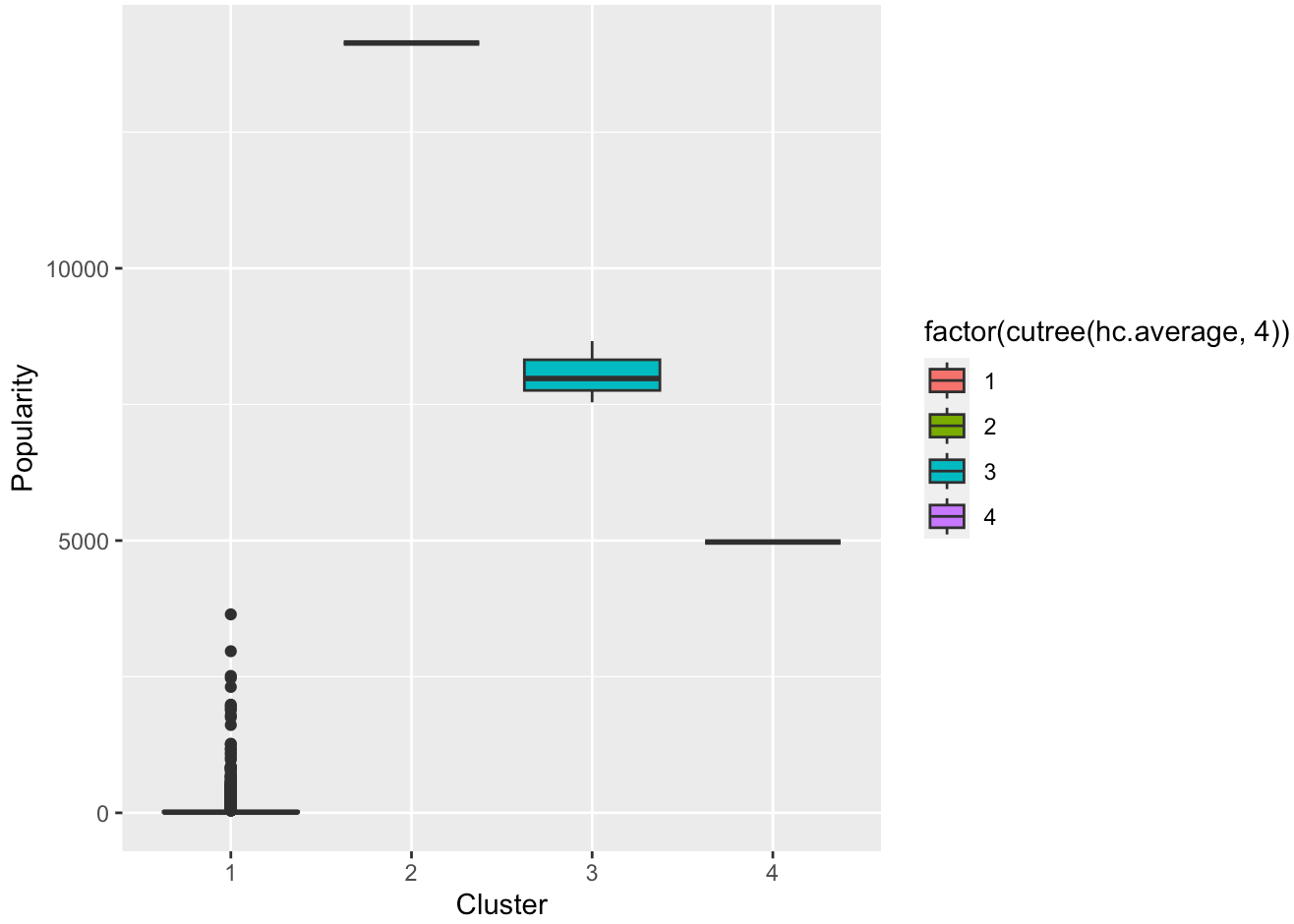

```
ggplot(movies, aes(x = factor(clusters), y = vote_average, fill = factor(clusters))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Vote Average")
```




```
ggplot(movies, aes(x = factor(cutree(hc.average,4)), y = vote_count, fill = factor(clust
ers))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Vote Count")
```




```
ggplot(movies, aes(x = factor(cutree(hc.average,4)), y = popularity, fill = factor(cutre
e(hc.average,4)))) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Popularity")
```


Popularity
10000
5000
0
1
2
3
4
Cluster
factor(cutree(hc.average, 4))
1
2
3
4