final_df_17_21_clustering_v2

Chen Lin, Xiaoyi Wang

2023-07-19

```
library(corrplot)
library(lubridate)
```

```
data <- read.csv("final_df_17_21.csv")

#View(data)
```

```
# head(data)
```

```
# Step 1: Convert to Date format
data$creation_date <- as.Date(data$creation_date)
data$last_access_date <- as.Date(data$last_access_date)

# Step 2: Calculate the difference between today and
today <- as.Date('2023-07-19')
date_diff1 <- difftime(today, data$creation_date, units = "days")
date_diff2 <- difftime(today, data$last_access_date, units = "days")

# Step 3: Create a new factor to store the calculated days
data$days_since_creation <- as.numeric(date_diff1)
data$days_since_last_access <- as.numeric(date_diff2)
```

```
# Specify the variables to drop
variables_to_drop <- c("id", "display_name", "location", "about_me",
                       "highest_scoring_question", "highest_scoring_answer",
                       "creation_date", "last_access_date", "_merge",
                       "account_age", "harmonic_mean", "ques_answer_cnt_avg",
                       "ques_score_avg", "ques_view_cnt_avg", "ans_score_avg",
                       "account_age_days", "score_difference",
                       "ques_median_score", "ans_median_score","X_merge",
                       "account_age_years", "year")

# Drop the variables from the dataset
data <- data[, setdiff(names(data), variables_to_drop)]
```

```
str(data)
```

```
## 'data.frame':    78103 obs. of  15 variables:
##  $ reputation_x                : int  1421 251590 5616 57082 3012 6004 39359 661 3742 23602 ...
##  $ user_upv                    : int  377 2348 660 1303 47 308 2891 21 1979 147 ...
```

```
##  $ user_downv                   : int  1 20 13 36 14 21 58 0 2 2 ...
##  $ user_views                   : int  96 19758 864 4560 116 334 4514 514 336 1169 ...
##  $ ques_cnt                     : int  9 159 24 10 26 25 36 21 64 44 ...
##  $ ques_answer_cnt_tot          : int  26 215 24 21 44 35 55 25 103 76 ...
##  $ ques_score_tot               : int  83 131 26 49 86 35 162 23 77 685 ...
##  $ ques_view_cnt_tot            : int  117954 88421 75303 26203 144613 34390 250050 26290 59504 8331
##  $ ans_cnt                      : int  1 1 1 2 3 1 1 3 1 1 ...
##  $ ans_score_tot                : int  1 1 0 2 0 2 2 2 0 1 ...
##  $ ques_score                   : num  72 16 10 11 61 14 100 5 14 524 ...
##  $ ans_score                    : num  1 1 0 2 0 2 2 1 0 1 ...
##  $ harmonic_mean_with_reputation: num  2803 473581 0 193201 0 ...
##  $ days_since_creation          : num  2528 5371 4554 4828 2624 ...
##  $ days_since_last_access       : num  302 297 427 298 307 313 299 608 904 301 ...
```

```
# head(data)
```

```
# View(data)
```

```
summary(data)
```

```
##   reputation_x       user_upv       user_downv         user_views
##  Min.   :     1   Min.   :    0   Min.   :    0.00   Min.   :     0.0
##  1st Qu.:   181   1st Qu.:   19   1st Qu.:    0.00   1st Qu.:    36.0
##  Median :   699   Median :   82   Median :    2.00   Median :   100.0
##  Mean   :  2638   Mean   :  330   Mean   :   22.58   Mean   :   304.4
##  3rd Qu.:  2052   3rd Qu.:  316   3rd Qu.:   10.00   3rd Qu.:   262.0
##  Max.   :485622   Max.   :29379   Max.   :25583.00   Max.   :131862.0
##     ques_cnt       ques_answer_cnt_tot ques_score_tot    ques_view_cnt_tot
##  Min.   :  1.000   Min.   :  0.000     Min.   :-25.000   Min.   :      6
##  1st Qu.:  1.000   1st Qu.:  1.000     1st Qu.:  0.000   1st Qu.:    377
##  Median :  2.000   Median :  2.000     Median :  1.000   Median :   1241
##  Mean   :  3.386   Mean   :  4.334     Mean   :  4.755   Mean   :   5201
##  3rd Qu.:  4.000   3rd Qu.:  5.000     3rd Qu.:  4.000   3rd Qu.:   3944
##  Max.   :229.000   Max.   :356.000     Max.   :917.000   Max.   :1587296
##     ans_cnt       ans_score_tot        ques_score        ans_score
##  Min.   : 1.000   Min.   : -27.000   Min.   :-11.000   Min.   : -12.000
##  1st Qu.: 1.000   1st Qu.:   0.000   1st Qu.:  0.000   1st Qu.:   0.000
##  Median : 1.000   Median :   0.000   Median :  1.000   Median :   0.000
##  Mean   : 1.516   Mean   :   2.143   Mean   :  3.333   Mean   :   1.976
##  3rd Qu.: 2.000   3rd Qu.:   2.000   3rd Qu.:  3.000   3rd Qu.:   1.000
##  Max.   :51.000   Max.   :1222.000   Max.   :765.000   Max.   :1222.000
##  harmonic_mean_with_reputation days_since_creation days_since_last_access
##  Min.   :-628764.0             Min.   :2027        Min.   : 297.0
##  1st Qu.:      0.0             1st Qu.:2692        1st Qu.: 299.0
##  Median :      0.0             Median :3386        Median : 306.0
##  Mean   :      Inf             Mean   :3424        Mean   : 523.3
##  3rd Qu.:    854.5             3rd Qu.:4053        3rd Qu.: 492.0
##  Max.   :      Inf             Max.   :5465        Max.   :2383.0
```

```
# Check for missing values in each variable
missing_values <- sapply(data, function(x) sum(is.na(x)))
```

```r
# Print the number of missing values in each variable
print(missing_values)
```

```
##                   reputation_x                       user_upv
##                              0                              0
##                     user_downv                     user_views
##                              0                              0
##                       ques_cnt            ques_answer_cnt_tot
##                              0                              0
##                 ques_score_tot              ques_view_cnt_tot
##                              0                              0
##                        ans_cnt                  ans_score_tot
##                              0                              0
##                     ques_score                      ans_score
##                              0                              0
## harmonic_mean_with_reputation           days_since_creation
##                              0                              0
##         days_since_last_access
##                              0
```

```r
# Check for infinite values
is_inf <- apply(data, 2, function(x) any(!is.finite(x)))
inf_vars <- names(is_inf)[is_inf]

# Print variables with infinite values
print(inf_vars)
```

```
## [1] "harmonic_mean_with_reputation"
```

```r
# Drop harmonic_mean_with_reputation
data <- subset(data, select = -harmonic_mean_with_reputation)
```

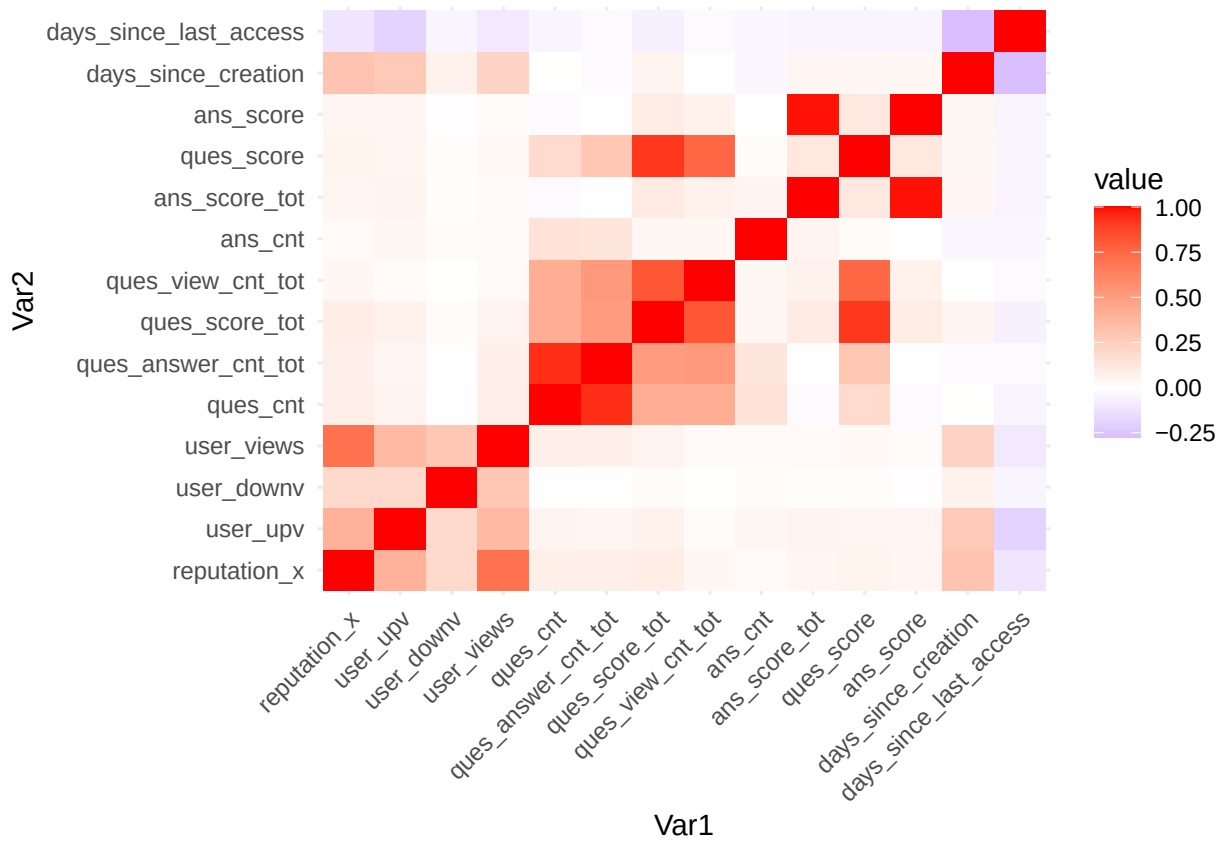
Scale the data

```r
# Scale the data
df = scale(data)
#View(df)
```

```r
# Example of correlation matrix heatmap using ggplot2
library(ggplot2)
library(reshape2)

# Assuming your data is stored in a data frame called 'data'
cor_matrix <- cor(df, method = "pearson") # Calculate the correlation matrix
melted_cor <- melt(cor_matrix)

ggplot(melted_cor, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
### PCA
library("ggplot2")
# library("FactoMineR")
# library("factoextra")
```

```
library("FactoMineR")

res.pca <- PCA(data, scale.unit = TRUE, graph = FALSE) # scale unit
print(res.pca)
```

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 78103 individuals, described by 14 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
```

```
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

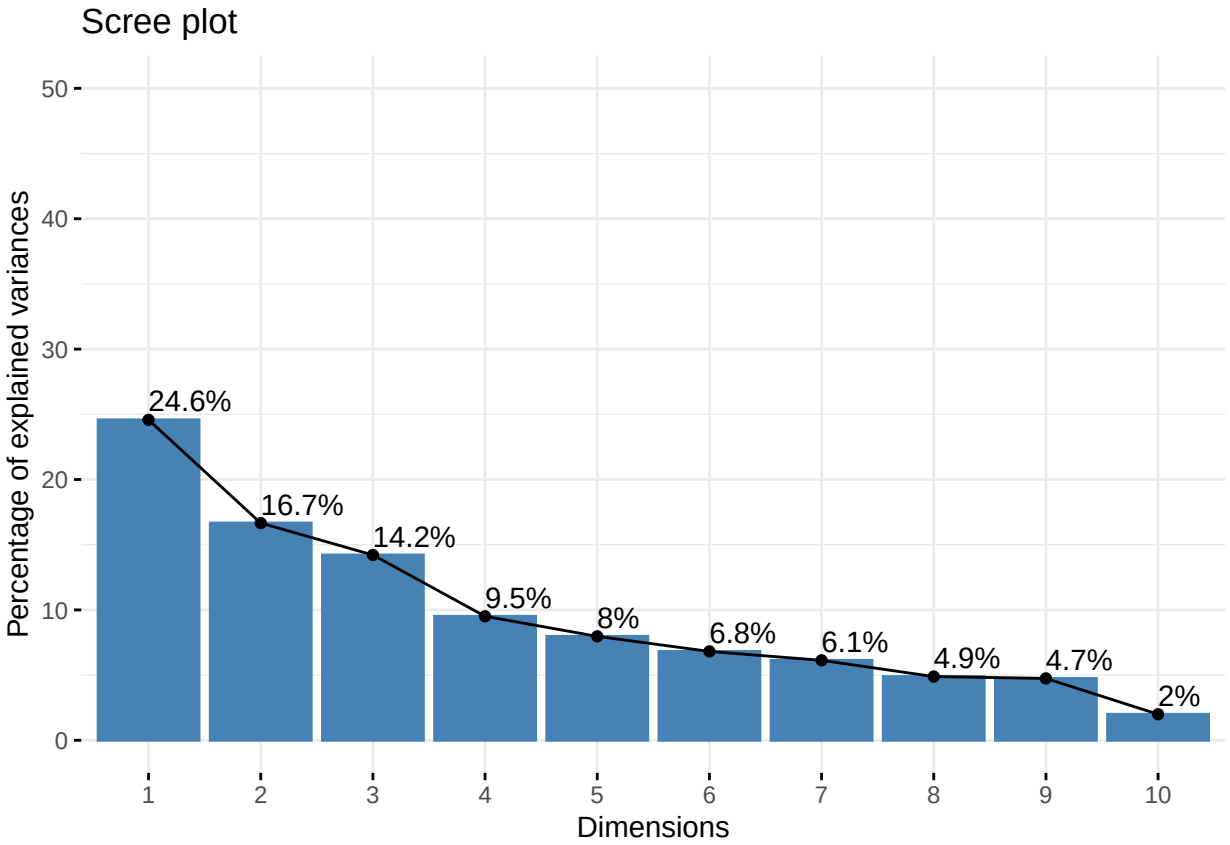
```
library("factoextra")
```

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

```
eig.val <- get_eigenvalue(res.pca)
eig.val
```

```
##        eigenvalue variance.percent cumulative.variance.percent
## Dim.1  3.44070162      24.57644016                    24.57644
## Dim.2  2.33248734      16.66062384                    41.23706
## Dim.3  1.98960171      14.21144077                    55.44850
## Dim.4  1.33141358       9.51009703                    64.95860
## Dim.5  1.11601915       7.97156538                    72.93017
## Dim.6  0.95417547       6.81553904                    79.74571
## Dim.7  0.85837058       6.13121841                    85.87692
## Dim.8  0.68405416       4.88610112                    90.76303
## Dim.9  0.66447840       4.74627428                    95.50930
## Dim.10 0.27954570       1.99675498                    97.50605
## Dim.11 0.23272506       1.66232189                    99.16838
## Dim.12 0.05442863       0.38877590                    99.55715
## Dim.13 0.05120085       0.36572034                    99.92287
## Dim.14 0.01079776       0.07712687                   100.00000
```

```
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0,50))
```



We want to stop at the eighth principal component. 41% of the information contained in the data are retained by the first eight principal components.

```
var <- get_pca_var(res.pca)
var
```

```
## Principal Component Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

```
# coordinates of variables
head(var$coord)
```

```
##                          Dim.1      Dim.2      Dim.3        Dim.4       Dim.5
## reputation_x         0.25431140  0.7577016 -0.1976011  0.005832766  0.19802148
## user_upv             0.19716760  0.6290462 -0.1300538 -0.019398463 -0.09215870
## user_downv           0.08681765  0.4049859 -0.1184889 -0.007618115  0.39768287
## user_views           0.22643078  0.7397250 -0.2311011  0.037810413  0.31218811
## ques_cnt             0.68176897 -0.1506811 -0.2584897  0.593671612 -0.10159977
## ques_answer_cnt_tot 0.75244794 -0.1853224 -0.2315612  0.515711067 -0.06539632
```

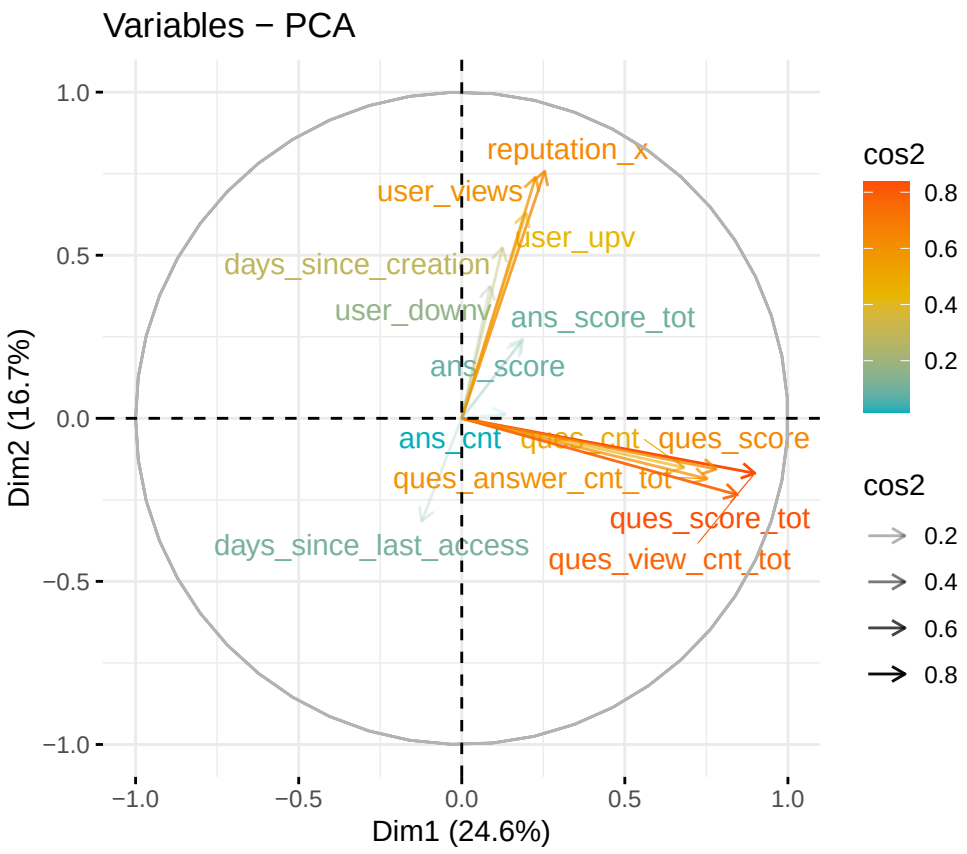
```r
# quality on the factor map
head(var$cos2)
```

```
##                           Dim.1      Dim.2      Dim.3        Dim.4       Dim.5
## reputation_x        0.064674287 0.57411165 0.03904621 3.402116e-05 0.039212507
## user_upv            0.038875061 0.39569907 0.01691400 3.763004e-04 0.008493226
## user_downv          0.007537304 0.16401354 0.01403962 5.803567e-05 0.158151666
## user_views          0.051270897 0.54719310 0.05340773 1.429627e-03 0.097461415
## ques_cnt            0.464808935 0.02270479 0.06681690 3.524460e-01 0.010322514
## ques_answer_cnt_tot 0.566177901 0.03434440 0.05362061 2.659579e-01 0.004276679
```

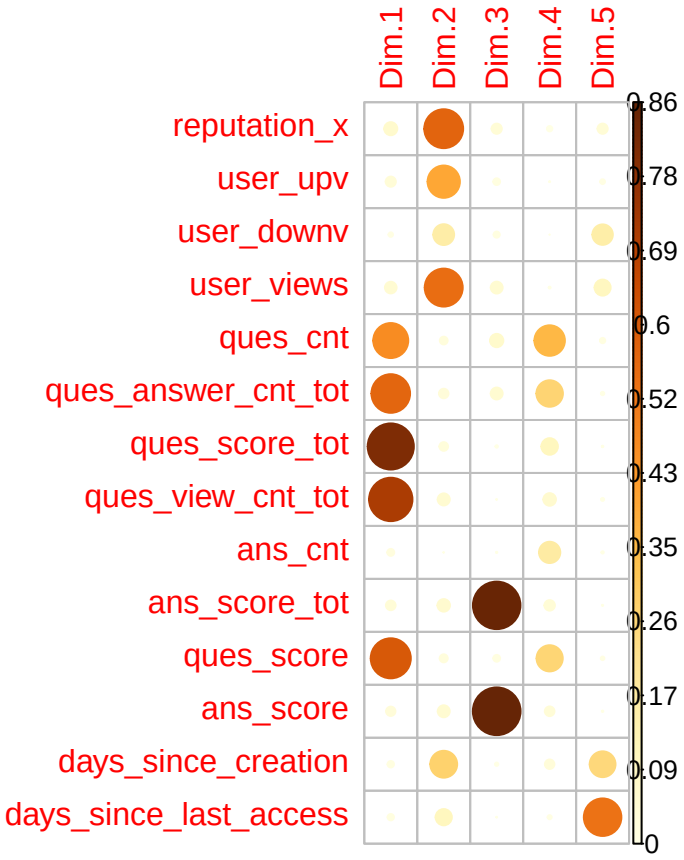
```r
# contribution of variables
head(var$contrib)
```

```
##                           Dim.1      Dim.2     Dim.3        Dim.4      Dim.5
## reputation_x          1.879683 24.6137094 1.9625140  0.002555266  3.5136052
## user_upv              1.129859 16.9646823 0.8501200  0.028263222  0.7610287
## user_downv            0.219063  7.0317013 0.7056498  0.004358952 14.1710531
## user_views            1.490129 23.4596385 2.6843426  0.107376653  8.7329518
## ques_cnt             13.509132  0.9734153 3.3583054 26.471562754  0.9249406
## ques_answer_cnt_tot 16.455304  1.4724365 2.6950422 19.975603977  0.3832083
```

```r
# color by cos2 values: quality on the factor map
fviz_pca_var(res.pca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE,
             alpha.var = "cos2"
)
```

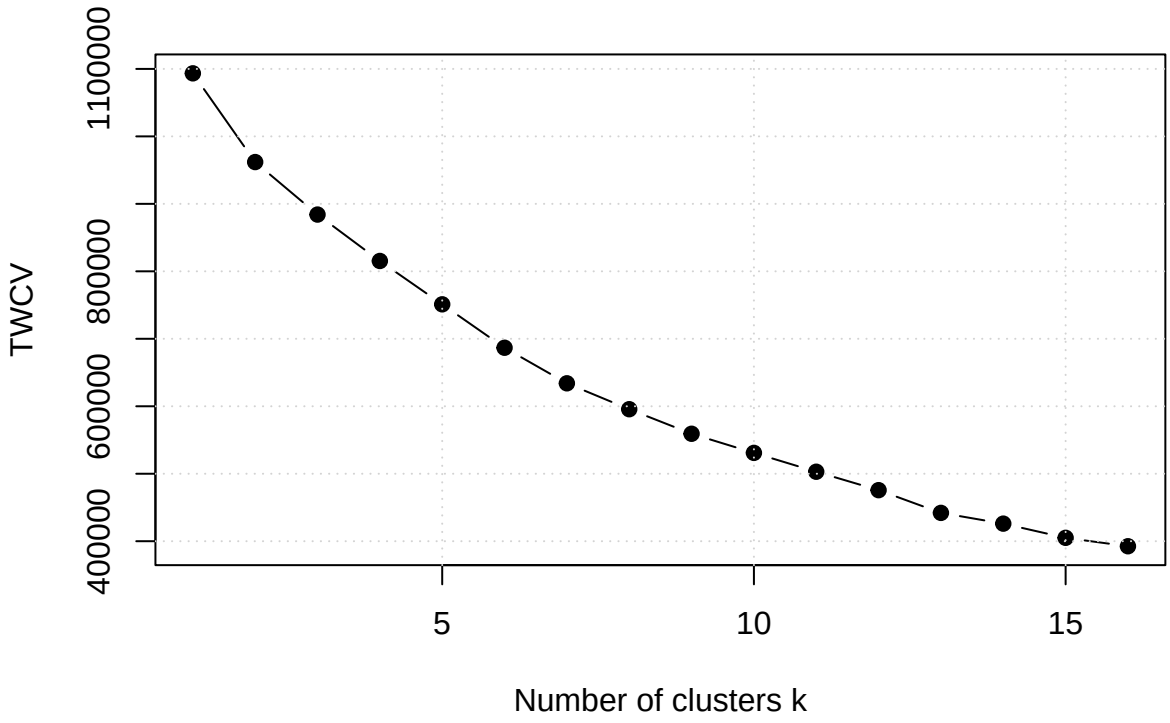



```
# visualize the cos2 of variables on all the dimensions
library("corrplot")
corrplot(var$cos2, is.corr = FALSE)
```



Kmeans Clustering with PCA

```
# Elbow chart
set.seed(123)
twcv = function(k) kmeans(df,k,nstart=25)$tot.withinss
#plot twcv
k = 1:16
twcv_values = sapply(k,twcv)
plot(k,twcv_values,type="b",pch=19,xlab="Number of clusters k",ylab="TWCV")
grid()
```

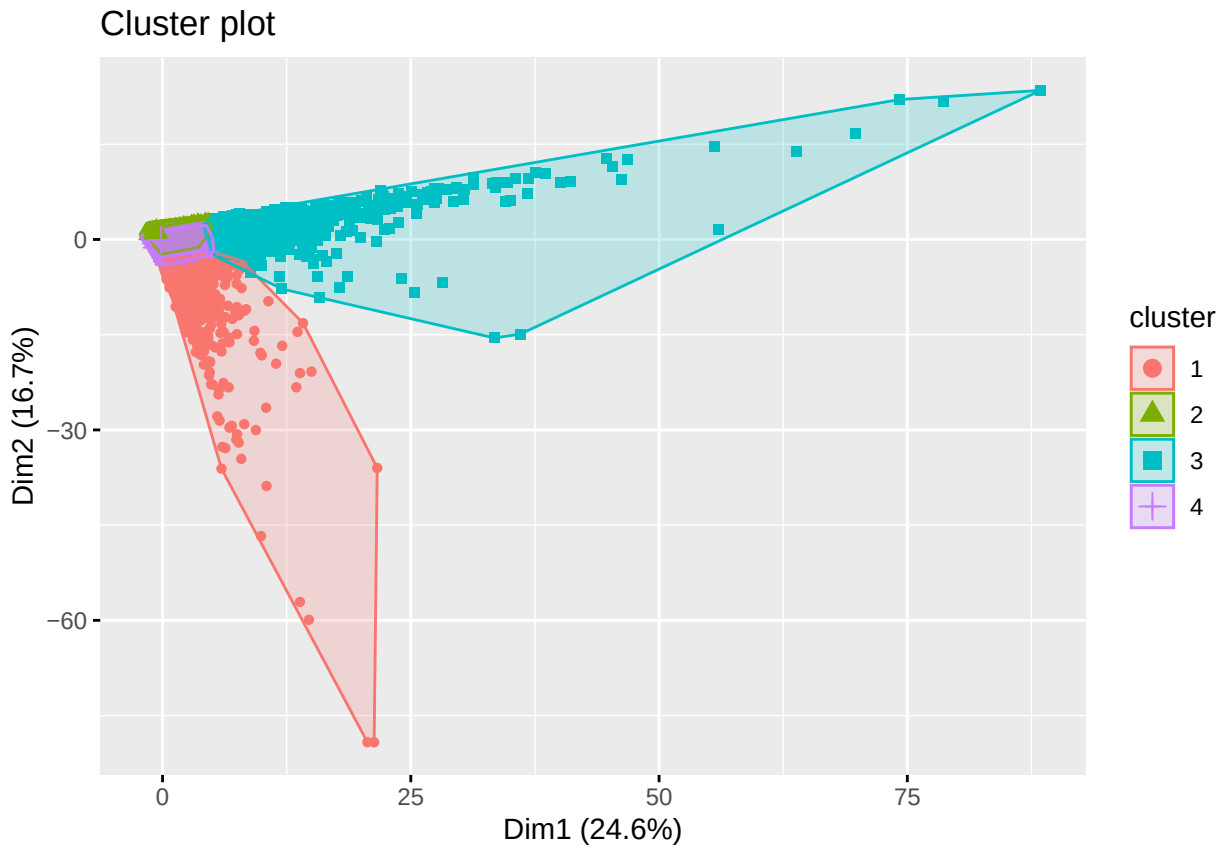



```r
set.seed(42)  # Set a seed for reproducibility

k = 4
kmeans_result <- kmeans(df, centers = k, nstart = 25)

# Access the cluster assignments
cluster_assignments <- kmeans_result$cluster

fviz_cluster(kmeans_result, geom = "point", data = df)
```



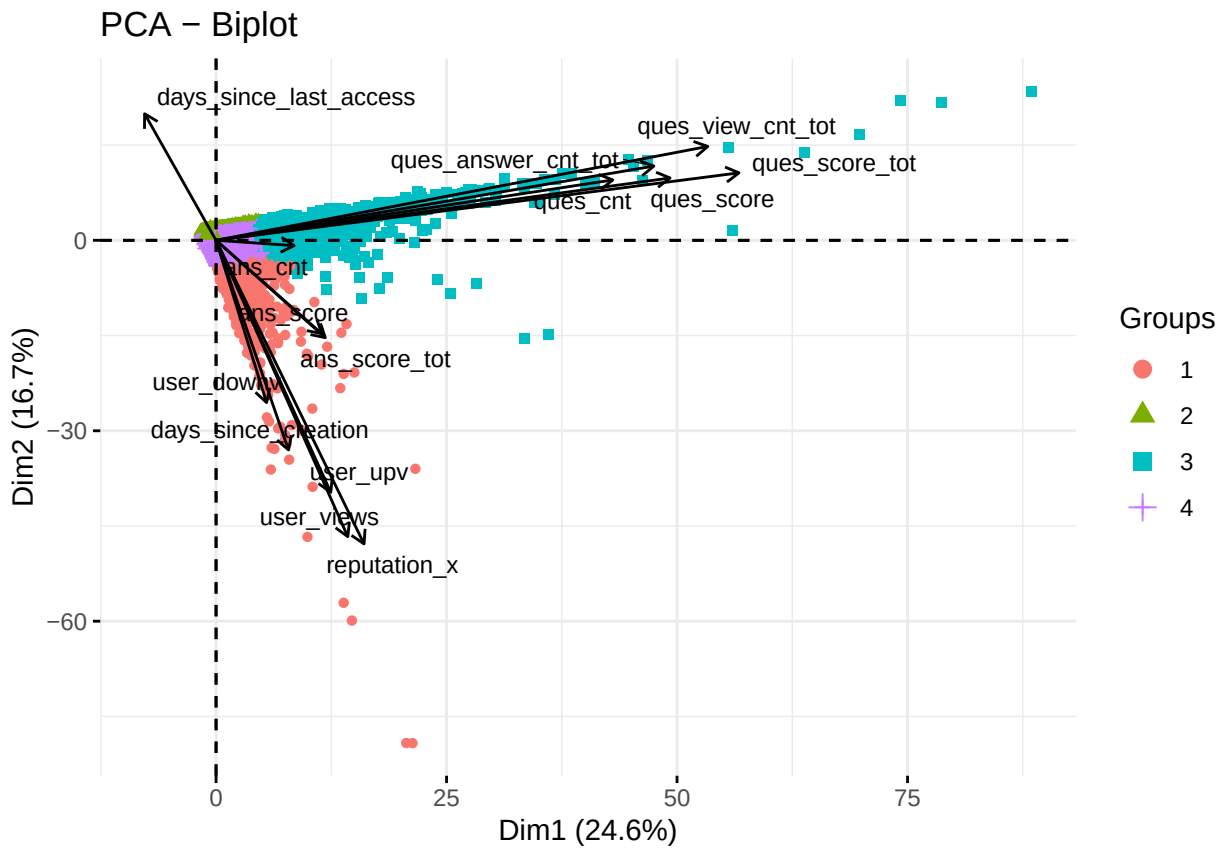

```
cluster_number = as.factor(kmeans_result$cluster)
data$cluster = cluster_number
head(data)
```

```
##   reputation_x user_upv user_downv user_views ques_cnt ques_answer_cnt_tot
## 1         1421      377          1         96        9                   26
## 2       251590     2348         20      19758      159                  215
## 3         5616      660         13        864       24                   24
## 4        57082     1303         36       4560       10                   21
## 5         3012       47         14        116       26                   44
## 6         6004      308         21        334       25                   35
##   ques_score_tot ques_view_cnt_tot ans_cnt ans_score_tot ques_score ans_score
## 1             83            117954       1             1         72         1
## 2            131             88421       1             1         16         1
## 3             26             75303       1             0         10         0
## 4             49             26203       2             2         11         2
## 5             86            144613       3             0         61         0
## 6             35             34390       1             2         14         2
##   days_since_creation days_since_last_access cluster
## 1                2528                    302       3
## 2                5371                    297       3
## 3                4554                    427       3
## 4                4828                    298       1
## 5                2624                    307       3
## 6                3718                    313       3
```

```
# View(data)
```

```
library(ggplot2)
library(factoextra)

# biplot with clusters
m1 = prcomp(df, scale=T)
fviz_pca_biplot(m1, geom = "point", col.var = "black",
  habillage = cluster_number, labelsize = 3, repel = TRUE)
```



Summary:

Group 1 - reputable contributors: reputable, active, long-time dedicated users. great contribution into building the community with high quality contents. willing to offer constructive feedback, by answering questions and voting to share their opinions and make impacts.

Group 2 - inactive users: low participation. they stopped making contribution.

Group 3 - curious learner: most active in raising questions. also willing to give it a try in answering questions as part of learning. (not necessarily giving perfect answers that receive high scores)

Group 4 - community builder: no specific preference in answering or asking questions.

```
print("Within cluster sum of squares by cluster:")
```

```
## [1] "Within cluster sum of squares by cluster:"
```

```
print(kmeans_result$betweenss/kmeans_result$totss)
```

```
## [1] 0.2543259
```

```
print("Size of each cluster:")
```

```
## [1] "Size of each cluster:"
```

```
print(kmeans_result$size)
```

```
## [1]  1422 10781  1399 64501
```

```
print(kmeans_result$centers)
```

```
##   reputation_x    user_upv  user_downv user_views    ques_cnt
## 1    4.2526239  3.89073874  2.02430996  3.4217984  0.02377832
## 2   -0.2574847 -0.41127423 -0.09998197 -0.2134349 -0.13631704
## 3    0.3505153  0.29666883  0.04318443  0.3165278  3.89294057
## 4   -0.0583194 -0.02346821 -0.02885348 -0.0466284 -0.06217582
##   ques_answer_cnt_tot ques_score_tot ques_view_cnt_tot      ans_cnt
## 1          0.01643852     0.17985939       -0.04885263  0.233996056
## 2         -0.10357015    -0.19078250       -0.10269392 -0.075482715
## 3          4.31705582     4.61269913        4.46269523  0.488652622
## 4         -0.07668636    -0.07212446       -0.07855226 -0.003140855
##   ans_score_tot  ques_score   ans_score days_since_creation
## 1    1.36805295  0.12995656  1.29770798          1.39512356
## 2   -0.11916209 -0.14965186 -0.10867753         -0.72373299
## 3    0.34372545  3.85078167  0.33518549          0.08546194
## 4   -0.01769828 -0.06137339 -0.01771465          0.08835736
##   days_since_last_access
## 1             -0.4500601
## 2              2.2049023
## 3             -0.2060631
## 4             -0.3541462
```