Data Mining Assignment 2

1) Sharad Deshmukh (PRN=2012) 2) Shubham Sonawane(PRN=2045)

22/02/2022

Introduction:

We have a given datasets availble in a file names as AppStore", which contain 17007 with with 14 variables. This dataset contains data on different mobile apps with given characteristics.

we are supposed to conduct different methods learnt so far. Given dataset has some values missing. So to analyse data further we need fill those empty spaces with so values using some method.

Data Preproccing

```
library(mice)
library(VIM)
```

```
AppStore <- read.csv("C:/Users/Sharad Deshmukh/Desktop/MSC=3/Data Mining/Data Minig Assignment/AppStore
s=duplicated.array(AppStore)
table(s)
```

```
## s
## FALSE  TRUE
## 16847   160
```

Here we get Complete 160 Rows Are Duplicate .

```
D=unique(AppStore)
dim(D)
```

```
## [1] 16847    13
```

```
summary(D)
```

```
##        ID                Name           Average.User.Rating User.Rating.Count
##  Min.   :2.849e+08   Length:16847       Min.   :1.000       Min.   :      5.0
##  1st Qu.:8.997e+08   Class :character   1st Qu.:3.500       1st Qu.:     12.0
##  Median :1.112e+09   Mode  :character   Median :4.500       Median :     46.0
##  Mean   :1.060e+09                      Mean   :4.062       Mean   :   3306.2
##  3rd Qu.:1.287e+09                      3rd Qu.:4.500       3rd Qu.:    307.2
##  Max.   :1.475e+09                      Max.   :5.000       Max.   :3032734.0
##                                         NA's   :9359        NA's   :9359
```

```
##      Price            Developer          Age.Rating         Languages
##  Min.   :  0.0000   Length:16847       Length:16847       Length:16847
##  1st Qu.:  0.0000   Class :character   Class :character   Class :character
##  Median :  0.0000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  0.8154
##  3rd Qu.:  0.0000
##  Max.   :179.9900
##  NA's   :24
##       Size            Primary.Genre         Genres
##  Min.   :5.133e+04   Length:16847       Length:16847
##  1st Qu.:2.295e+07   Class :character   Class :character
##  Median :5.674e+07   Mode  :character   Mode  :character
##  Mean   :1.158e+08
##  3rd Qu.:1.330e+08
##  Max.   :4.006e+09
##  NA's   :1
##  Original.Release.Date Current.Version.Release.Date
##  Length:16847          Length:16847
##  Class :character      Class :character
##  Mode  :character      Mode  :character
##
##
##
##
```

Summary gives us the missing values for each variables which in last NA form.

```
str(D)
```

```
## 'data.frame':    16847 obs. of  13 variables:
##  $ ID                          : int  284921427 284926400 284946595 285755462 285831220 286210009 28
##  $ Name                        : chr  "Sudoku" "Reversi" "Morocco" "Sudoku (Free)" ...
##  $ Average.User.Rating         : num  4 3.5 3 3.5 3.5 3 2.5 2.5 2.5 2.5 ...
##  $ User.Rating.Count           : int  3553 284 8376 190394 28 47 35 125 44 184 ...
##  $ Price                       : num  2.99 1.99 0 0 2.99 0 0 0.99 0 0 ...
##  $ Developer                   : chr  "Mighty Mighty Good Games" "Kiss The Machine" "Bayou Games" "M
##  $ Age.Rating                  : chr  "4+" "4+" "4+" "4+" ...
##  $ Languages                   : chr  "DA, NL, EN, FI, FR, DE, IT, JA, KO, NB, PL, PT, RU, ZH, ES, S
##  $ Size                        : num  15853568 12328960 674816 21552128 34689024 ...
##  $ Primary.Genre               : chr  "Games" "Games" "Games" "Games" ...
##  $ Genres                      : chr  "Games, Strategy, Puzzle" "Games, Strategy, Board" "Games, Boa
##  $ Original.Release.Date       : chr  "11-07-2008" "11-07-2008" "11-07-2008" "23-07-2008" ...
##  $ Current.Version.Release.Date: chr  "30-05-2017" "17-05-2018" "05-09-2017" "30-05-2017" ...
```

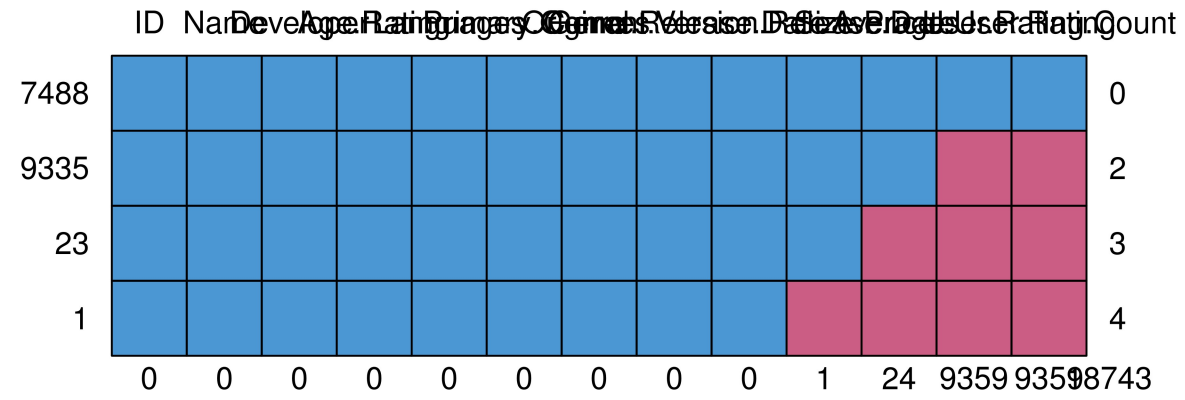
```
p <- function(x) {sum(is.na(x))/length(x)*100}
apply(D, 2, p)  ##Percentage of missing values
```

```
##                           ID                         Name
##                  0.000000000                  0.000000000
##          Average.User.Rating            User.Rating.Count
##                 55.552917433                 55.552917433
##                        Price                    Developer
##                  0.142458598                  0.000000000
```

```
##                   Age.Rating                    Languages
##                  0.000000000                  0.000000000
##                         Size                Primary.Genre
##                  0.005935775                  0.000000000
##                       Genres         Original.Release.Date
##                  0.000000000                  0.000000000
## Current.Version.Release.Date
##                  0.000000000
```

It gives percentage of miussing values in each variables

```
md.pattern(D)
```



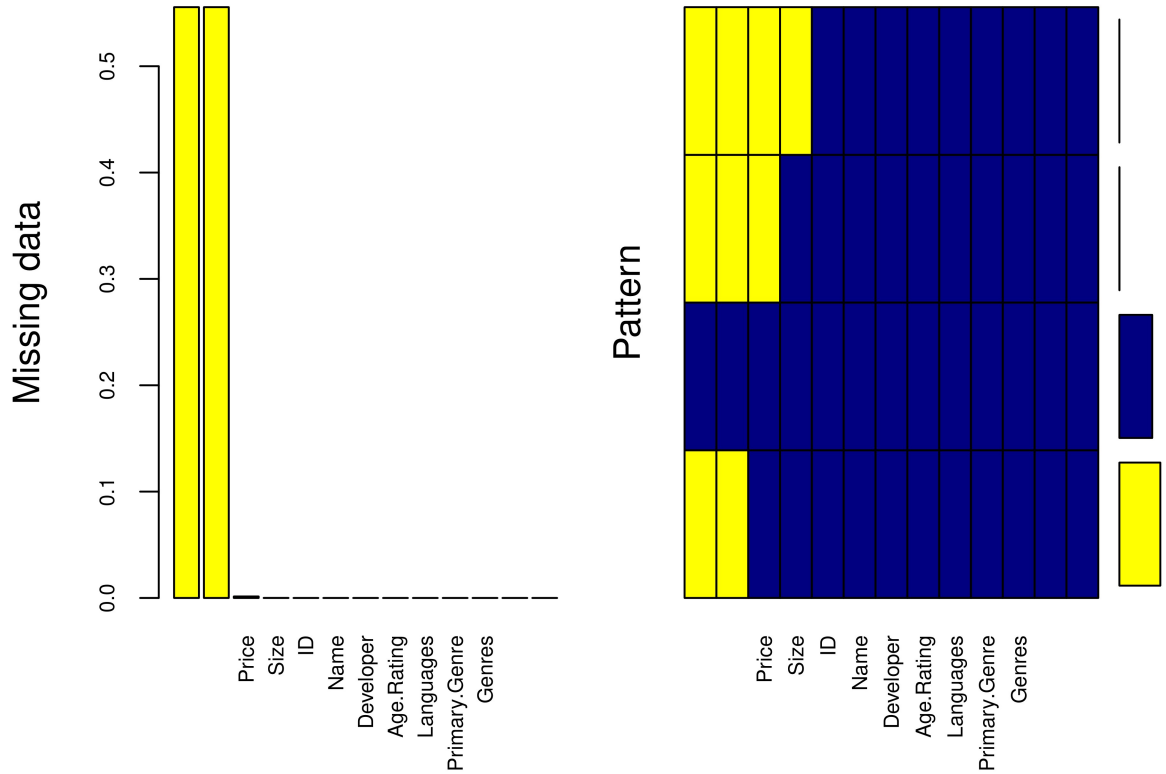

```
##      ID Name Developer Age.Rating Languages Primary.Genre Genres
## 7488  1    1         1          1         1             1      1
## 9335  1    1         1          1         1             1      1
## 23    1    1         1          1         1             1      1
## 1     1    1         1          1         1             1      1
##       0    0         0          0         0             0      0
##      Original.Release.Date Current.Version.Release.Date Size Price
## 7488                     1                            1    1     1
## 9335                     1                            1    1     1
## 23                       1                            1    1     0
## 1                        1                            1    0     0
##                          0                            0    1    24
```

```
##      Average.User.Rating User.Rating.Count
## 7488                   1                 1     0
## 9335                   0                 0     2
## 23                     0                 0     3
## 1                      0                 0     4
##                     9359              9359 18743
```

red cells represents the variable for which the row contains missing value and corresponsing no of on left represents the no of rows for which varibles having red cells contains missing values.

```
mice_plot <- aggr(D, col=c('navyblue','yellow'),
numbers=TRUE, sortVars=TRUE,
labels=names(D), cex.axis=.7,
gap=3, ylab=c("Missing data","Pattern"))  ##VAriables sorted by missing
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Variables sorted by number of missings:
##                       Variable        Count
##           Average.User.Rating 5.555292e-01
##             User.Rating.Count 5.555292e-01
##                         Price 1.424586e-03
```

```
##                          Size 5.935775e-05
##                            ID 0.000000e+00
##                          Name 0.000000e+00
##                     Developer 0.000000e+00
##                    Age.Rating 0.000000e+00
##                     Languages 0.000000e+00
##                  Primary.Genre 0.000000e+00
##                         Genres 0.000000e+00
##          Original.Release.Date 0.000000e+00
##   Current.Version.Release.Date 0.000000e+00
```

```
##So missing values variables are  Average.User.Rating ,User.Rating.Count ,Price,size
```

Above plot also gives us the Missing value Percentage.

```
##Only 1 value missing in "size" variables so used simple mean mputation
D$Size[which(is.na(D$Size))]=mean(D$Size,na.rm=TRUE)
summary(D)
```

```
##        ID                  Name           Average.User.Rating User.Rating.Count
##  Min.   :2.849e+08   Length:16847       Min.   :1.000       Min.   :      5.0
##  1st Qu.:8.997e+08   Class :character   1st Qu.:3.500       1st Qu.:     12.0
##  Median :1.112e+09   Mode  :character   Median :4.500       Median :     46.0
##  Mean   :1.060e+09                      Mean   :4.062       Mean   :   3306.2
##  3rd Qu.:1.287e+09                      3rd Qu.:4.500       3rd Qu.:    307.2
##  Max.   :1.475e+09                      Max.   :5.000       Max.   :3032734.0
##                                         NA's   :9359        NA's   :9359
##      Price            Developer          Age.Rating         Languages
##  Min.   :  0.0000   Length:16847       Length:16847       Length:16847
##  1st Qu.:  0.0000   Class :character   Class :character   Class :character
##  Median :  0.0000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  0.8154
##  3rd Qu.:  0.0000
##  Max.   :179.9900
##  NA's   :24
##       Size            Primary.Genre         Genres
##  Min.   :5.133e+04   Length:16847       Length:16847
##  1st Qu.:2.295e+07   Class :character   Class :character
##  Median :5.675e+07   Mode  :character   Mode  :character
##  Mean   :1.158e+08
##  3rd Qu.:1.330e+08
##  Max.   :4.006e+09
##
##  Original.Release.Date Current.Version.Release.Date
##  Length:16847          Length:16847
##  Class :character      Class :character
##  Mode  :character      Mode  :character
##
##
##
##
```

Only 1 value missing in “size” variables so used simple mean imputation

```
##so remiaing 3 imputation

##Mice inputation
impute <- mice(D[,c(3,4,5)],method ="pmm", m=5, seed = 123)
```

```
##
##  iter imp variable
##   1   1  Average.User.Rating  User.Rating.Count  Price
##   1   2  Average.User.Rating  User.Rating.Count  Price
##   1   3  Average.User.Rating  User.Rating.Count  Price
##   1   4  Average.User.Rating  User.Rating.Count  Price
##   1   5  Average.User.Rating  User.Rating.Count  Price
##   2   1  Average.User.Rating  User.Rating.Count  Price
##   2   2  Average.User.Rating  User.Rating.Count  Price
##   2   3  Average.User.Rating  User.Rating.Count  Price
##   2   4  Average.User.Rating  User.Rating.Count  Price
##   2   5  Average.User.Rating  User.Rating.Count  Price
##   3   1  Average.User.Rating  User.Rating.Count  Price
##   3   2  Average.User.Rating  User.Rating.Count  Price
##   3   3  Average.User.Rating  User.Rating.Count  Price
##   3   4  Average.User.Rating  User.Rating.Count  Price
##   3   5  Average.User.Rating  User.Rating.Count  Price
##   4   1  Average.User.Rating  User.Rating.Count  Price
##   4   2  Average.User.Rating  User.Rating.Count  Price
##   4   3  Average.User.Rating  User.Rating.Count  Price
##   4   4  Average.User.Rating  User.Rating.Count  Price
##   4   5  Average.User.Rating  User.Rating.Count  Price
##   5   1  Average.User.Rating  User.Rating.Count  Price
##   5   2  Average.User.Rating  User.Rating.Count  Price
##   5   3  Average.User.Rating  User.Rating.Count  Price
##   5   4  Average.User.Rating  User.Rating.Count  Price
##   5   5  Average.User.Rating  User.Rating.Count  Price
```

```
summary(impute)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
## Average.User.Rating   User.Rating.Count               Price
##               "pmm"               "pmm"               "pmm"
## PredictorMatrix:
##                      Average.User.Rating User.Rating.Count Price
## Average.User.Rating                    0                 1     1
## User.Rating.Count                      1                 0     1
## Price                                  1                 1     0
```

Here we fit the missing values Pnn method which is called predictive mean matching.miceRanger can make use of a procedure called predictive mean matching (PMM) to select which values are imputed. PMM involves selecting a datapoint from the original, nonmissing data which has a predicted value close to the predicted value of the missing sample.

```
##clean data

head(impute$imp$Average.User.Rating) ##5 imputed values choose any of 5 for missing values
```

```
##       1   2   3   4   5
## 11  3.5 4.5 4.5 4.5 4.5
## 23  4.0 3.5 4.5 4.5 4.5
## 24  4.0 4.5 4.0 5.0 5.0
## 57  4.0 4.5 4.0 5.0 4.0
## 70  4.0 5.0 4.0 4.5 5.0
## 166 4.0 5.0 4.5 4.5 5.0
```

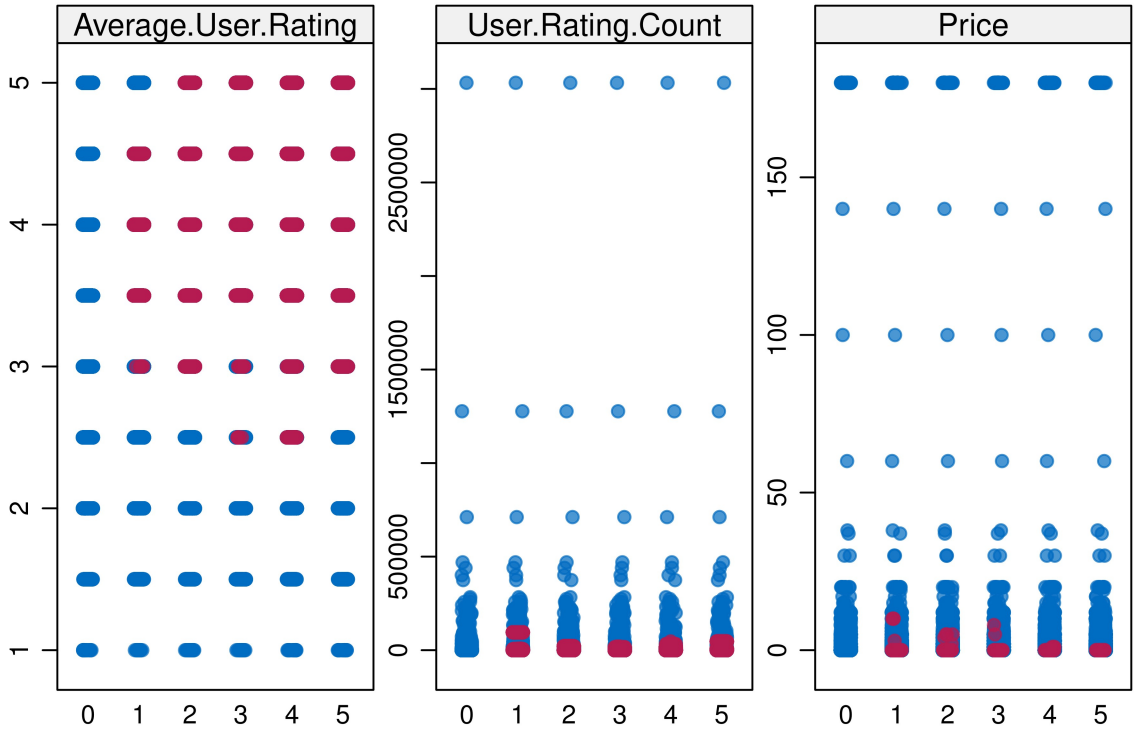
This are the 5 important imputed values from this we had choose column 3 number impute with missing values.

```
CD=complete(impute,3)  ##choose 3
summary(CD)
```

```
##  Average.User.Rating User.Rating.Count     Price
##  Min.   :1.000       Min.   :      5   Min.   :  0.000
##  1st Qu.:4.000       1st Qu.:     19   1st Qu.:  0.000
##  Median :4.500       Median :     55   Median :  0.000
##  Mean   :4.263       Mean   :   2240   Mean   :  0.815
##  3rd Qu.:4.500       3rd Qu.:    458   3rd Qu.:  0.000
##  Max.   :5.000       Max.   :3032734   Max.   :179.990
```

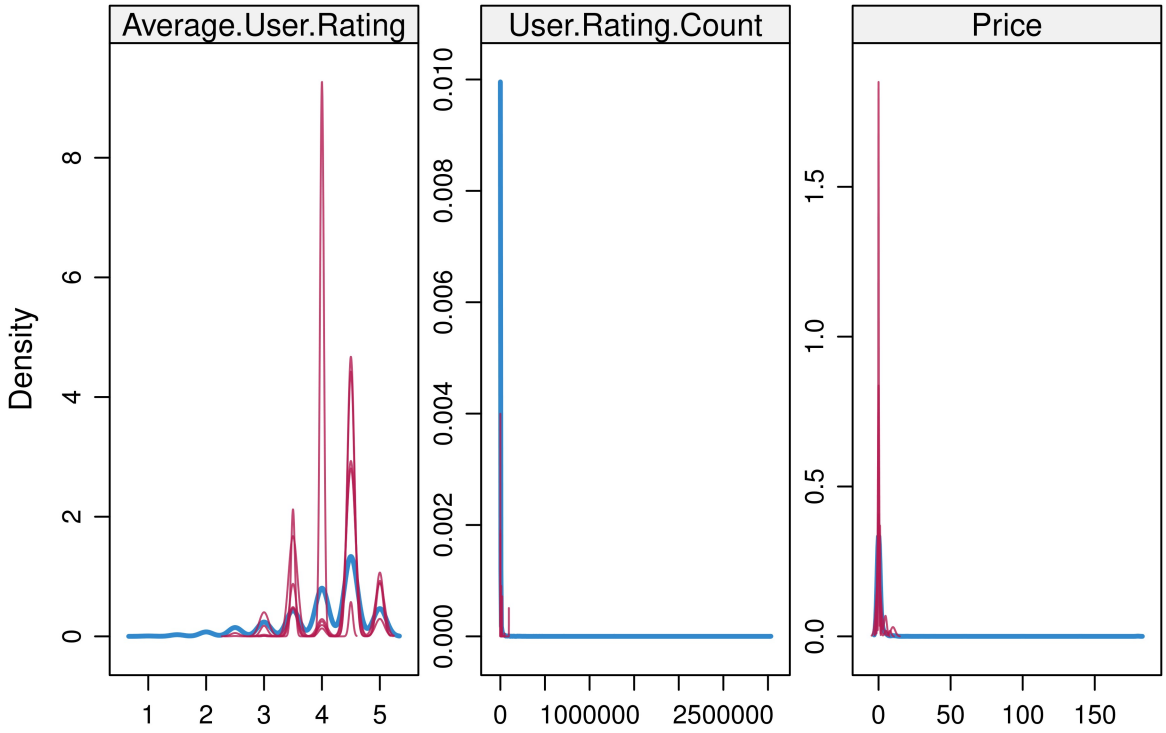
from this table we Fit all missing values.

```
##Plot observed and imputed data
##
stripplot(impute,pch=20,cex=1.2)   ##Red are missing data and blue are original data
```



Red are missing data and blue are original. there is good pattern of missing data that we can see from this plot.

```
densityplot(impute) ##red lines are 5 imputed data and blue line is original date
```



so this is the plot of density plot of 5 imputed data that we generated from mice command .Red line is for missing imputed data set.basically in imputation they try to bootsrap the sample after they try to used different method like pmm etc.so we have 5 imputated data.blue line is for original data.so we can see that it fills well for missing data.

```
D[,c(3,4,5)]=CD[,c(1,2,3)]  ##imputatation of missing values
summary(D)
```

```
##        ID                  Name           Average.User.Rating User.Rating.Count
##  Min.   :2.849e+08   Length:16847       Min.   :1.000       Min.   :      5
##  1st Qu.:8.997e+08   Class :character   1st Qu.:4.000       1st Qu.:     19
##  Median :1.112e+09   Mode  :character   Median :4.500       Median :     55
##  Mean   :1.060e+09                      Mean   :4.263       Mean   :   2240
##  3rd Qu.:1.287e+09                      3rd Qu.:4.500       3rd Qu.:    458
##  Max.   :1.475e+09                      Max.   :5.000       Max.   :3032734
##      Price          Developer          Age.Rating         Languages
##  Min.   :  0.000   Length:16847       Length:16847       Length:16847
##  1st Qu.:  0.000   Class :character   Class :character   Class :character
##  Median :  0.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  0.815
##  3rd Qu.:  0.000
##  Max.   :179.990
##       Size            Primary.Genre         Genres
##  Min.   :5.133e+04   Length:16847       Length:16847
##  1st Qu.:2.295e+07   Class :character   Class :character
##  Median :5.675e+07   Mode  :character   Mode  :character
```

```
##  Mean   :1.158e+08
##  3rd Qu.:1.330e+08
##  Max.   :4.006e+09
##  Original.Release.Date Current.Version.Release.Date
##  Length:16847          Length:16847
##  Class :character      Class :character
##  Mode  :character      Mode  :character
##
##
##
```

SO Fill all missing values from original data as you can see from summary able.

Q1. Create some sort of Popularity Index and rank the apps as per their popularity.Clearly mention the variables that you have used and explain your reasoning behind

the choice of variables and the choice of the index (formula for the index)

.

```
##1)
D$Average.User.Rating*D$User.Rating.Count->index
Popularity=data.frame(D,index)
D1=Popularity[order(-Popularity$index),]
head(D1)
```

```
##               ID                            Name Average.User.Rating
## 1379   529479190                  Clash of Clans                 4.5
## 7188  1053012308                    Clash Royale                 4.5
## 13415 1330123889                     PUBG MOBILE                 4.5
## 1922   597986893    Plants vs. Zombies\\u2122 2                 4.5
## 12474 1270598321 Cash, Inc. Fame & Fortune Game                 5.0
## 2411   672150402                      Boom Beach                 4.5
##       User.Rating.Count Price                              Developer Age.Rating
## 1379            3032734     0                              Supercell         9+
## 7188            1277095     0                              Supercell         9+
## 13415            711409     0 Tencent Mobile International Limited        17+
## 1922             469562     0                                 PopCap         9+
## 12474            374772     0                           Lion Studios         4+
## 2411             400787     0                              Supercell         9+
##                                                                                Languages
## 1379  AR, NL, EN, FI, FR, DE, ID, IT, JA, KO, MS, NB, PT, RU, ZH, ES, TH, ZH, TR, VI
## 7188                      AR, NL, EN, FR, DE, IT, JA, KO, NB, PT, RU, ZH, ES, ZH, TR
## 13415                                ZH, EN, FR, DE, ID, PT, RU, ZH, ES, TH, ZH, TR
## 1922                                                    EN, FR, DE, IT, PT, ES
## 12474                                                                          EN
## 2411          AR, NL, EN, FR, DE, ID, IT, JA, KO, MS, NB, PT, RU, ZH, ES, ZH, TR, VI
##              Size Primary.Genre                                    Genres
## 1379   161219584         Games     Games, Action, Entertainment, Strategy
## 7188   145107968         Games     Games, Strategy, Entertainment, Action
## 13415 2384081920         Games                   Games, Action, Strategy
```

```
## 1922   120763392         Games  Games, Strategy, Entertainment, Adventure
## 12474  245957632         Games Games, Strategy, Entertainment, Simulation
## 2411   202785792         Games                    Games, Strategy, Action
##       Original.Release.Date Current.Version.Release.Date    index
## 1379             02-08-2012                  20-06-2019 13647303
## 7188             02-03-2016                  01-08-2019  5746928
## 13415            19-03-2018                  12-06-2019  3201341
## 1922             15-08-2013                  29-07-2019  2113029
## 12474            06-10-2017                  12-07-2019  1873860
## 2411             26-03-2014                  03-07-2019  1803542
```

In this question we are supposed to create some popularity index to rank the apps as per their popularity. For that we chosen the variables Average User Rating

and User Rating Count. The former gives the average rating per user and the later tells the total no of user ratings so far. If we take the product of these two the it ###### will kind of give total rating received so far and that would work sort of good index for the popularity.

Q.2. If an app that I like turns out to be a paid app, how can I locate a similar free app?

```
###Sudoko##2)
sd=which(D$Genres=="Games, Strategy, Puzzle" )  ##we analyze genres
length(sd)
```

```
## [1] 768
```

```
price=D$Price[sd]                ##genres prices
names=D$Name[sd]                 ##Names of Genres
z=data.frame(names,price)
unpaid=which(z$price==0.00)     ##unpaid prices
Z=z[unpaid,]
head(Z)
```

```
##             names price
## 2   Sudoku (Free)     0
## 3       Gaia Lite     0
## 5  RoboLogic Lite     0
## 6      Flip Ninja     0
## 9      Pipe Mania     0
## 11   Strimko Lite     0
```

if the app somebody like turns out to be the paid app then we can locate the similar app using the variable which are more likely to be the reason for choosing that particular game. Looking at the dataset and looking at the peoples playing games nowadays we can say that geners could be the more probable variable for choosing that particular variable.

so for this we have choosen a particular that somebody supposed to like and then we extracted all those games with their prices which have similar genres and then from this particular dataframe we extracted all those games which are free their are almost ... such games.

Q3) Rank the genres (using both the genre related columns) as per their popularity in

18+ age group. Explain how you use the app popularity to determine the genre popularity.

```
##3)
adult=which(D1$Age.Rating=="17+")
Q=D1[adult,]
P=Q[1:10,c(2,10,11)];P  ##Top 10 Games
```

```
##                                    Name Primary.Genre
## 13415                       PUBG MOBILE         Games
## 8389  South Park: Phone Destroyer\\u2122         Games
## 156         Yahoo Fantasy Football & more        Sports
## 11723       Mafia City: War of Underworld         Games
## 11054           "Wiz Khalifa's Weed Farm"         Games
## 11612                Jurassic World Alive         Games
## 664                           Crime City         Games
## 883                           Modern War         Games
## 5580       Walking Dead: Road to Survival         Games
## 8264                Bud Farm: Grass Roots         Games
##                                          Genres
## 13415                     Games, Action, Strategy
## 8389                        Games, Card, Strategy
## 156               Sports, Games, Sports, Strategy
## 11723                           Games, Strategy
## 11054          Games, Simulation, Strategy, Music
## 11612                  Games, Strategy, Adventure
## 664   Games, Entertainment, Role Playing, Strategy
## 883                  Games, Strategy, Role Playing
## 5580  Games, Role Playing, Strategy, Entertainment
## 8264                  Games, Strategy, Simulation
```

We ranked the genres as per the ranking of corresponding apps having 17+ age group as age rating and red cells represents the variable for which the row contains missing value and corresponsing no of on left represents the no of rows for which varibles having red cells contains missing values.

```
unique(Q$Primary.Genre)
```

```
##  [1] "Games"             "Sports"            "Education"
##  [4] "Entertainment"     "Lifestyle"         "Finance"
##  [7] "Utilities"         "News"              "Business"
## [10] "Social Networking" "Book"              "Music"
## [13] "Reference"
```

```
length(which(Q$Primary.Genre=="Games"))  ##Games
```

```
## [1] 611
```

```
length(which(Q$Primary.Genre== "Sports")) ##Sports
```

```
## [1] 16
```

From most of top 10 application primary genre as Games also as you can see from the length of 611 games that shows most of app has games genre

Q4)Using the given data, report one most interesting finding (i.e., frame your own question and answer it) other than those used in the earlier questions.

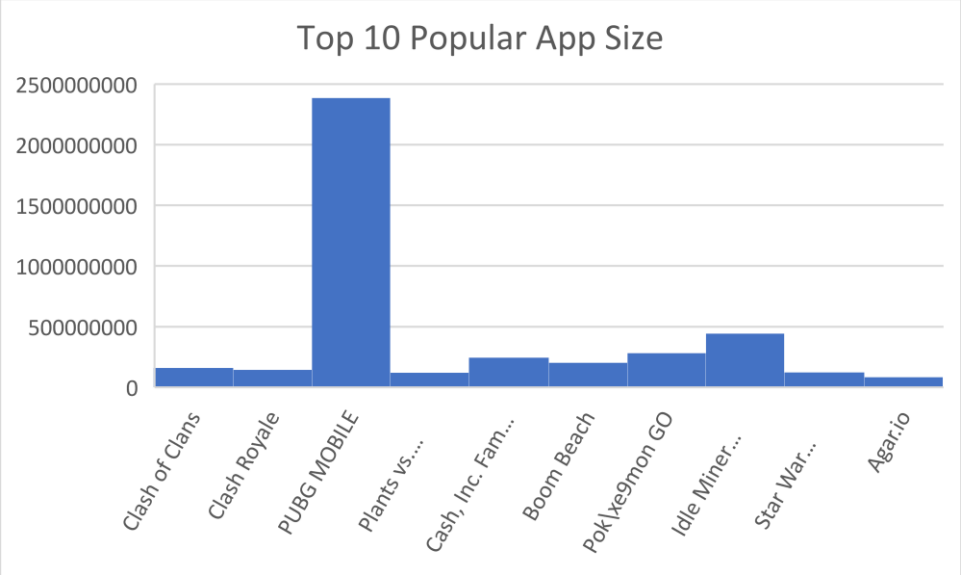
```
D1[1:10,c(2,3,4,5,6,7,8,9,10,11)]  ##Popularity app
```

```
##                                Name Average.User.Rating User.Rating.Count
## 1379                 Clash of Clans                 4.5           3032734
## 7188                   Clash Royale                 4.5           1277095
## 13415                   PUBG MOBILE                 4.5            711409
## 1922     Plants vs. Zombies\\u2122 2                 4.5            469562
## 12474 Cash, Inc. Fame & Fortune Game                 5.0            374772
## 2411                     Boom Beach                 4.5            400787
## 8140                  Pok\\xe9mon GO                 3.5            439776
## 8633  Idle Miner Tycoon: Cash Empire                 4.5            283035
## 3551     Star Wars\\u2122: Commander                 4.5            259030
## 6145                         Agar.io                 4.5            257852
##       Price                            Developer Age.Rating
## 1379      0                            Supercell         9+
## 7188      0                            Supercell         9+
## 13415     0 Tencent Mobile International Limited        17+
## 1922      0                              PopCap         9+
## 12474     0                        Lion Studios         4+
## 2411      0                            Supercell         9+
## 8140      0                        Niantic, Inc.         9+
## 8633      0                   Kolibri Games GmbH         4+
## 3551      0                        NaturalMotion         9+
## 6145      0                         Miniclip.com         9+
##
## 1379                                  AR, NL, EN, FI, FR, DE, ID, IT, JA, KO, MS, NB, PT, RU, ZH, ES
## 7188                                                  AR, NL, EN, FR, DE, IT, JA, KO, NB, PT, RU
## 13415                                                    ZH, EN, FR, DE, ID, PT, RU, ZH
## 1922                                                                                  EN, FR
## 12474
## 2411                                        AR, NL, EN, FR, DE, ID, IT, JA, KO, MS, NB, PT, RU, ZH
## 8140                                                                        EN, FR, DE, IT, JA
## 8633  ZH, CS, DA, NL, EN, FI, FR, DE, HI, HU, ID, IT, JA, KO, MS, NB, PL, PT, RO, RU, ZH, ES, SV, TH
## 3551                                                                  EN, FR, DE, IT, JA, KO, PT
## 6145                                                                        EN, FR, DE, IT, JA
##             Size Primary.Genre                                        Genres
## 1379   161219584         Games          Games, Action, Entertainment, Strategy
## 7188   145107968         Games          Games, Strategy, Entertainment, Action
## 13415 2384081920         Games                         Games, Action, Strategy
## 1922   120763392         Games       Games, Strategy, Entertainment, Adventure
```

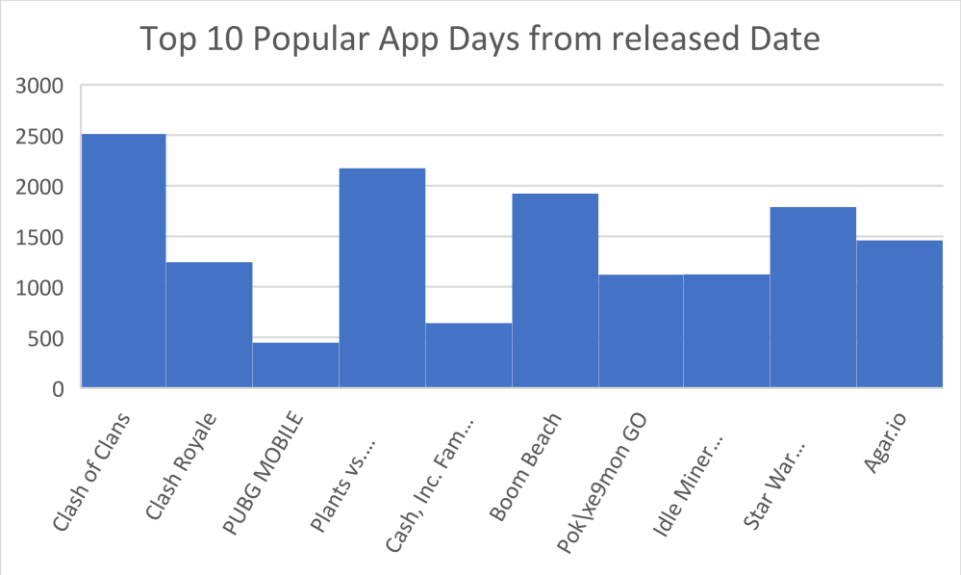
```
## 12474  245957632         Games      Games, Strategy, Entertainment, Simulation
## 2411   202785792         Games                          Games, Strategy, Action
## 8140   281521152         Games Games, Strategy, Role Playing, Health & Fitness
## 8633   443974656         Games      Games, Simulation, Strategy, Entertainment
## 3551   123083776         Games          Games, Entertainment, Action, Strategy
## 6145    83948544         Games          Games, Strategy, Entertainment, Action
```

Finding 1.From the top 10 most popular apps we can see that most of them are available in many languages.And that's obvious because apps that are more popula are supposed to be available in many languages

Finding 2. The apps that are at the top are very large in size this is because there popularity will be based on graphics and facilities that they contain



From Above Histogram we get that Pubg is the highest in size from in compare to other top 10 App.



From Above Histogram We can see that the pubg is latest among the top 10 apps but still it is most popular

We see that Class Of clan is the oldest latest among the top 10 apps but still it is most popular which shows the consistency.