Final Project MATH 208

Shuo Wang

12/12/2021

Environment setting

```
library(magrittr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
library(ggplot2)
library(patchwork)
library(rpart)
library(rpart.plot)
```

Data set loading

```
df<- read.csv("Final_Project_FlixGem.csv")
```

TASK ONE

A.

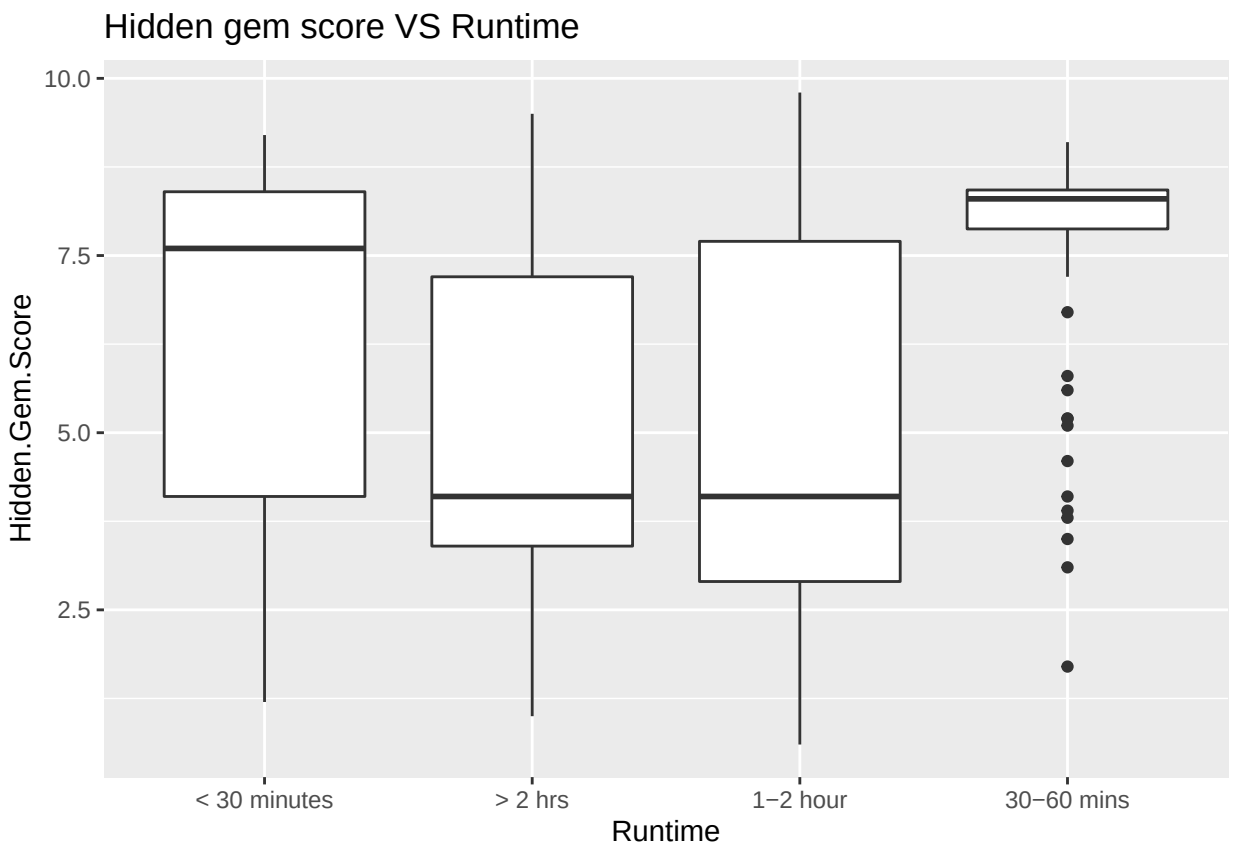
Data Cleaning

```
#delet NA values of specific variables
#only keep movies, delet series
#To ensure the accuracy of result, I drop NA values of each variable pair respectively
HGS_Rt<-df %>% drop_na(c(Hidden.Gem.Score,Runtime)) %>% filter(Series.or.Movie=='Movie')

HGS_Lan<-df %>% drop_na(c(Hidden.Gem.Score,Languages)) %>% filter(Series.or.Movie=='Movie')
```

Plot to visualize the relationship between variables

```
#From the plot below, movies runtime with 30-60 mins earn the highest score(above 7.5)
#movies less than 30 mins performs well with around 7.5
#movies longer than 1 hour, earned score less than 5.0 near 3.25.
#note that there are no movie longer than 2 hour with score 7.5 or better.
p<-ggplot(data=HGS_Rt,
       mapping = aes(
         x=Runtime,
         y=Hidden.Gem.Score
         )
       )+geom_boxplot()+labs(title="Hidden gem score VS Runtime")
print(p)
```




```
#We pick the top ten languages to do the next analysis
a<-HGS_Lan%>% count(Languages,sort=TRUE)
head(a)
```

```
##          Languages    n
## 1          English 2801
## 2         Japanese  446
## 3 English, Spanish  222
## 4           Korean  186
## 5            Hindi  168
## 6  English, French  127
```

```
aa<-a%>% top_n(10) #use the top 10 languages to analyse
```

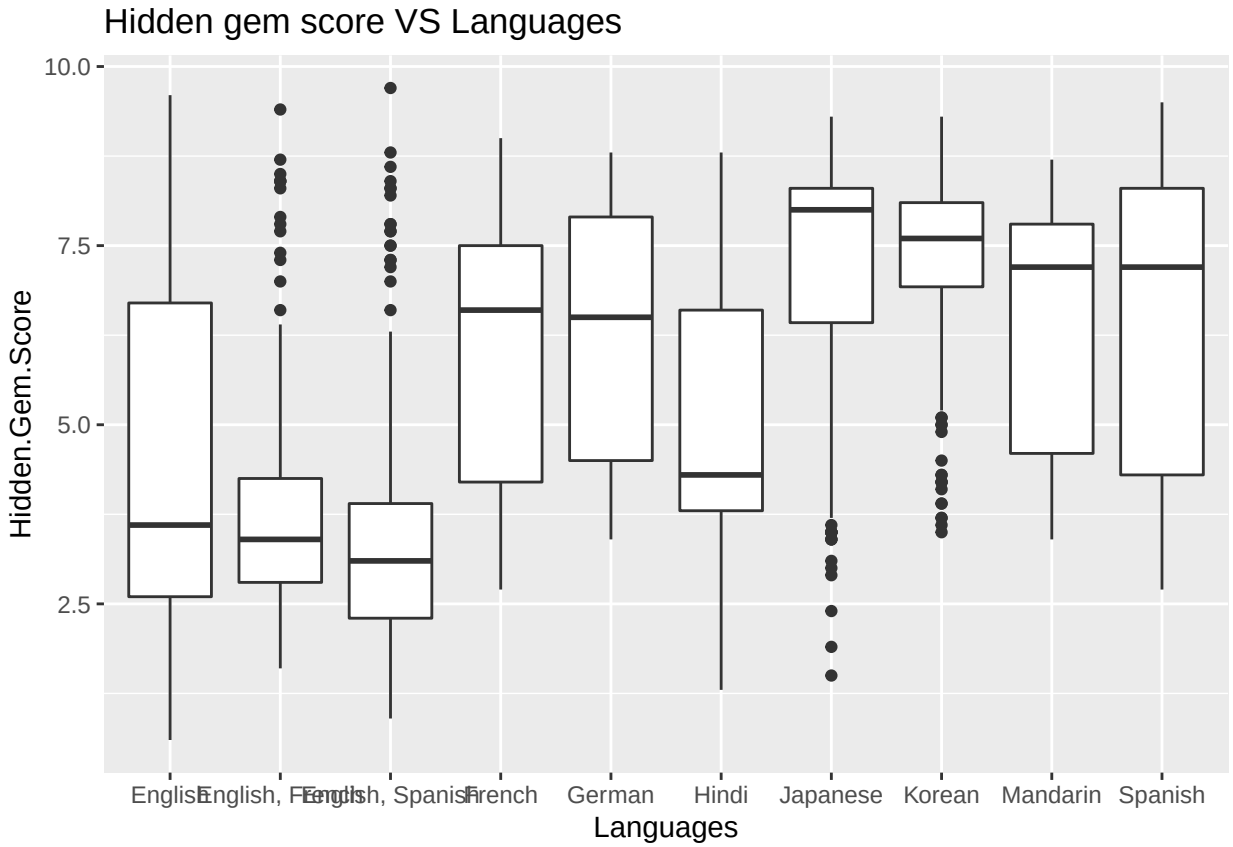
```
## Selecting by n
```

```
b<-as.vector(aa$Languages)
c<-HGS_Lan %>%filter(Languages %in% b) %>% summarize(Hidden.Gem.Score,Languages)
head(c)
```

```
##   Hidden.Gem.Score Languages
## 1              7.0   English
## 2              2.0   English
## 3              8.8   Spanish
## 4              3.5   English
## 5              4.4   English
## 6              3.8    French
```

From the box-plot of top ten languages (includes multi-language) , we can see that the Asian(Japanese,Korean,Chinese) movies have better score, in particular, Japanese has the highest median(8.0) and most Korean movies distribute within the interval (6.5,8).English movies have lowest score most below 4.0.

```
ggplot(data=c,
       mapping = aes(
         x=Languages,
         y=Hidden.Gem.Score
         )
       )+geom_boxplot()+labs(title="Hidden gem score VS Languages")
```



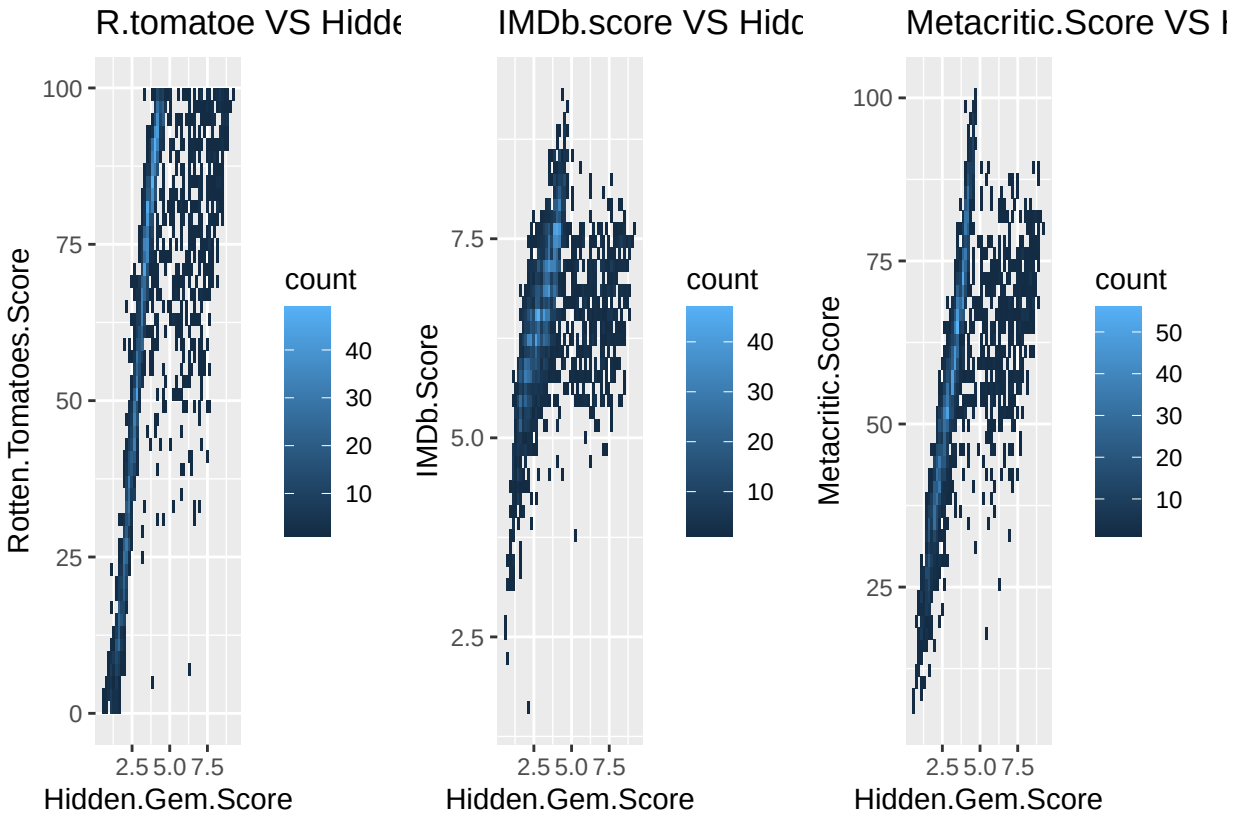
B

```
#data cleaning
reviews<-df%>% drop_na(c(Hidden.Gem.Score,IMDb.Score,Rotten.Tomatoes.Score,Metacritic.Score))%>% filter
head(reviews)
```

```
##   Hidden.Gem.Score IMDb.Score Rotten.Tomatoes.Score Metacritic.Score
## 1              7.0        5.8                    79               69
## 2              2.0        6.2                    20               36
## 3              3.5        8.4                    68               59
## 4              2.8        6.5                    52               51
## 5              4.4        8.1                    96               85
## 6              3.8        7.0                    85               72
```

Use2-D histogram to find if there are correlation between review site scores and Hidden Gem score.

```
p1<-ggplot(reviews,aes(x=Hidden.Gem.Score,y=Rotten.Tomatoes.Score))+stat_bin2d(bins=50)+ggtitle("R.toma
p2<-ggplot(reviews,aes(x=Hidden.Gem.Score,y=IMDb.Score))+stat_bin2d(bins=50)+ggtitle("IMDb.score VS Hid
p3<-ggplot(reviews,aes(x=Hidden.Gem.Score,y=Metacritic.Score))+stat_bin2d(bins=50)+ggtitle("Metacritic.
p1+p2+p3
```



conclusion: The tree plots all show the linear correlation between scores. Hidden.Gem.Score strongly has correlation to both Rotten.Tomatoes and Metacritic scores. IMDb.score not that strongly as the other two, but still clearly demonstrate the correlation especially in higher score part.

C

By the Hidden gem score VS Runtime Plot, it draw conclusions contrary to company's theory, that is people prefer the 30-60 mins films rather than longer length movies.

TASK TWO

pre_code dataset

```
dtset<-df%>% filter(Series.or.Movie=='Movie',Languages %in% b) %>% select(Hidden.Gem.Score,Languages,Ru
head(dtset)
```

```
##   Hidden.Gem.Score Languages  Runtime IMDb.Score Metacritic.Score
## 1              7.0   English 1-2 hour        5.8               69
## 2              2.0   English 1-2 hour        6.2               36
## 3              3.5   English 1-2 hour        8.4               59
## 4              4.4   English 1-2 hour        8.1               85
## 5              3.8    French 1-2 hour        7.0               72
## 6              6.2   English  > 2 hrs        6.9               51
##   Rotten.Tomatoes.Score
```

```
## 1                    79
## 2                    20
## 3                    68
## 4                    96
## 5                    85
## 6                    61
```

set trainset and testset

```
set.seed(514)
ind=sample(2,nrow(dtset),replace = T,prob = c(0.8,0.2))
trainset<-dtset[ind==1,]
testset<-dtset[ind==2,]

#dimension of trianset and testset
dim(trainset)
```

```
## [1] 1878    6
```

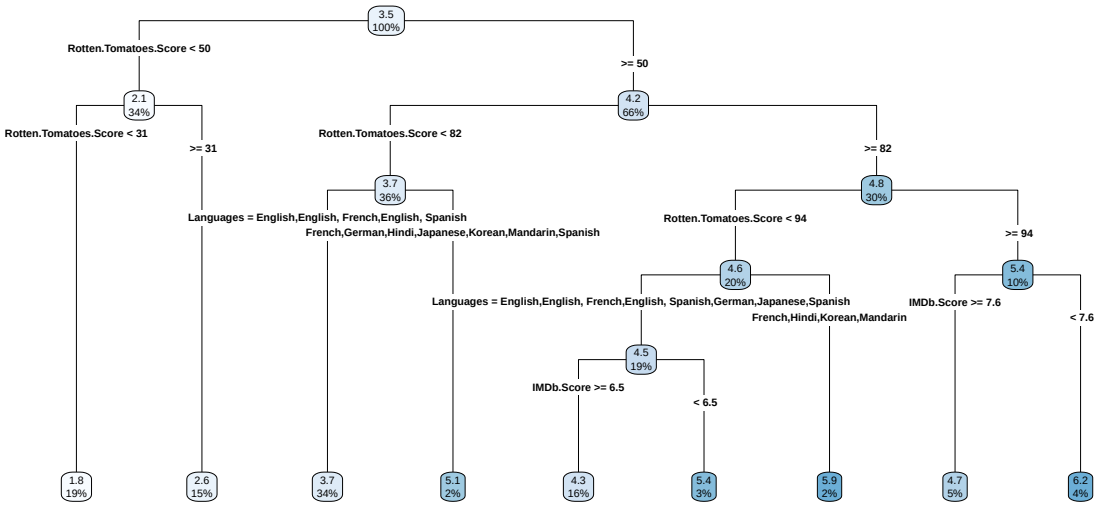
```
dim(testset)
```

```
## [1] 448   6
```

use rpart function generate a regression tree model

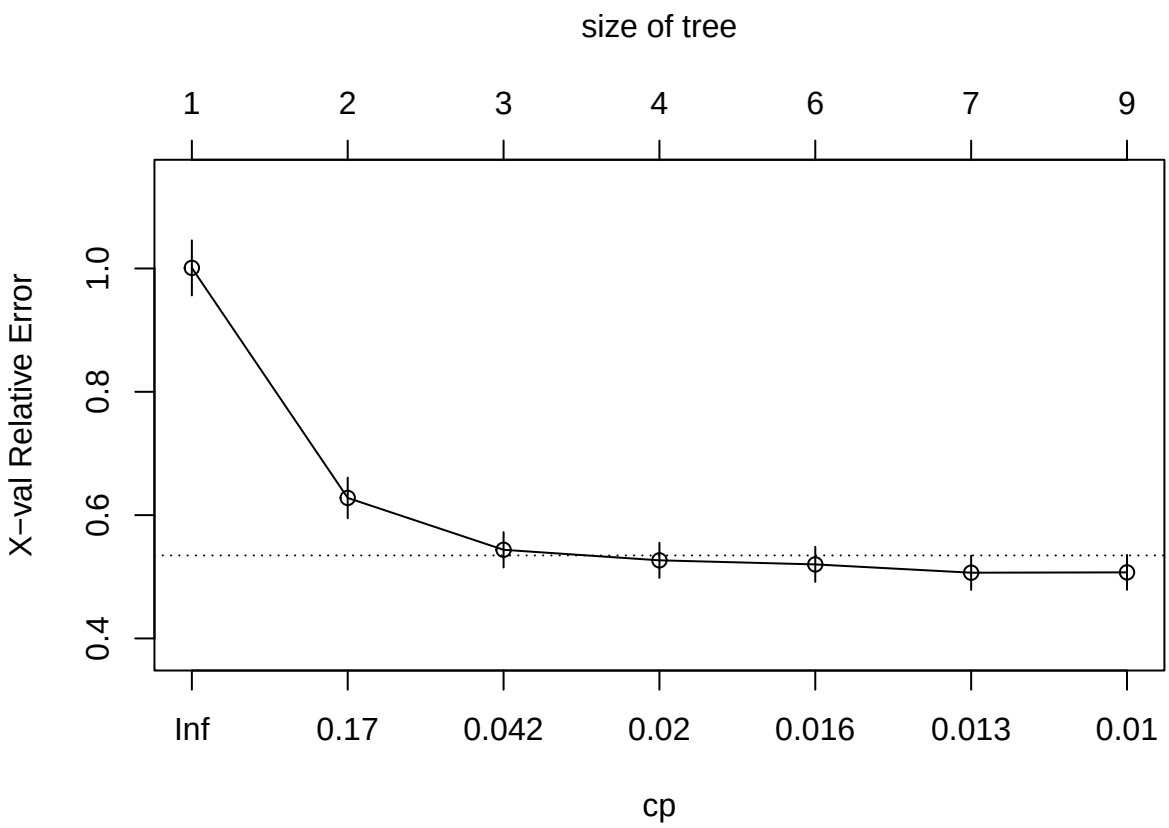
```
tree<-rpart(Hidden.Gem.Score~.,data=trainset,method = "anova")

rpart.plot(tree,4)
```



visualize cross validation error vs cost complexity value

```
plotcp(tree)
```



find the node of the min record of a cross-validated error

```
which.min(tree$cptable[,"xerror"])
```

```
## 6
## 6
```

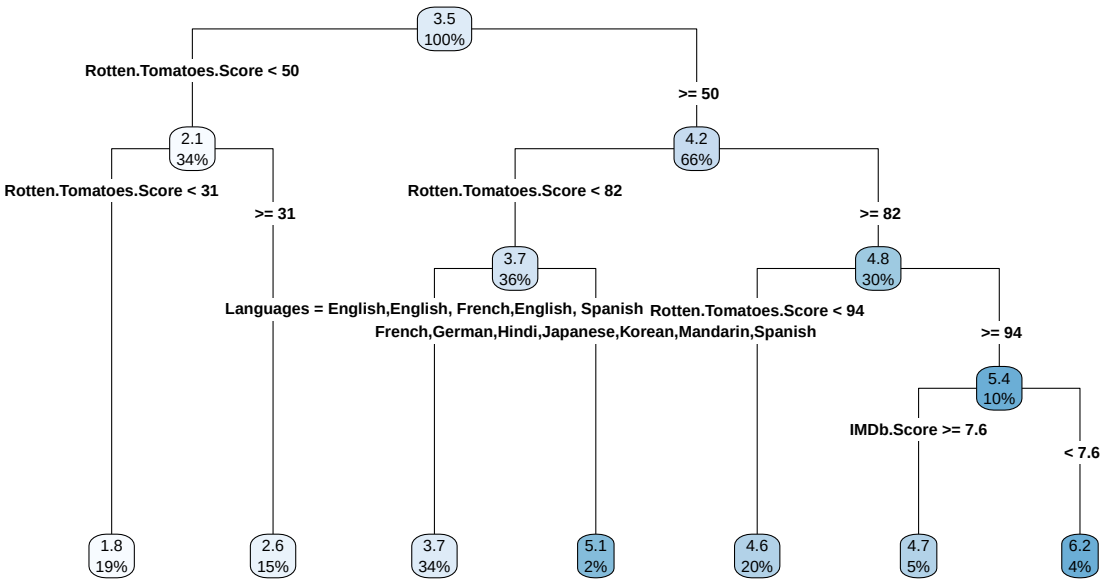
get cp value of the min xerror

```
tree.cp<-tree$cptable[6,"CP"]
tree.cp
```

```
## [1] 0.01084375
```

model performance improvement(prune tree)

```
prune.tree<-prune(tree,cp=tree.cp)
rpart.plot(prune.tree,4)
```



<u>Conclusion</u>: the most important factor is ROTTEN TOMATOES SCORE, languages as second, IMDb score comes third.

the minimum cross validation error is 0.5066283

```
prune.tree$cptable
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.38012937      0 1.0000000 1.0009785 0.04465008
## 2 0.07496350      1 0.6198706 0.6278914 0.03290715
## 3 0.02319349      2 0.5449071 0.5437550 0.02865210
## 4 0.01668433      3 0.5217136 0.5266617 0.02850576
## 5 0.01445745      5 0.4883450 0.5200766 0.02851390
## 6 0.01084375      6 0.4738875 0.5066283 0.02798450
```

TASK THREE

Dataset Recoding

```
df1<-df%>% filter(Series.or.Movie=='Movie')%>% drop_na() %>% group_by(Director) %>% summarise(Hidden.Ge
head(df1)
```

```
## # A tibble: 6 x 3
##   Director                     Hidden.Gem.Score Num_Movies
##   <chr>                                   <dbl>      <int>
## 1 Aaron Katz                                6            1
## 2 Aaron Lieber                              8.1          1
## 3 Aaron Sorkin                              3.8          1
## 4 Abby Kohn, Marc Silverstein               2.3          1
## 5 Abdellatif Kechiche                       4.2          1
## 6 Adam B. Stein, Zach Lipovsky              3.6          1
```

```
nrow(df1)#find how many rows in df1
```

```
## [1] 2118
```

Algorithm of H-index

By comparing each score with number of movies from a director, get the number of TURE results if match then the number is the H-index, if not continue to compare with n-1 number of movies until match.

step 0

```
# to speed up the process we first remove the director with only 1 and 2 movie.
# but this step is optional
df2<-df1 %>% filter(Num_Movies!=1&Num_Movies!=2)
nrow(df2)#number of rows after optimization
```

```
## [1] 1026
```

```
director_names<-as.vector(unique(df2$Director))
n=length(director_names)
dtf<-tibble("directors"=director_names,"H_index"="plog_in values later")
```

step 1 :create a new tibble with directors names

```
for (i in 1:n){
    tag<-dtf$directors[i] #chose the ith name in the director name list
    df3<-df2 %>% filter(Director==tag)
    HS<-df3$Hidden.Gem.Score
    NM<-df3$Num_Movies
    N=NM[1]#total movies

    for(j in 0:N){
    #cat("i is",i,"j is ",j,"N-j is",N-j)

    result<-(HS>=(N-j))

    nTrue<-sum(result) #find how many movies scores >= movies numbers
   # cat("N-j is---",N-j,"----Ntrue is",nTrue,"===")
    if(nTrue>=(N-j)){
      H_Index<-N-j
      dtf$H_index[i]<-H_Index
      break
```

```
      }
  }
  }
head(dtf,n=20)
```

step2: calculate the H-index of each directors by using for loops

```
## # A tibble: 20 x 2
##    directors             H_index
##    <chr>                 <chr>
##  1 Adam McKay            3
##  2 Adam Wingard          2
##  3 Adrian Lyne           2
##  4 Alejandro G. Iñárritu 3
##  5 Alexander Payne       2
##  6 Alfonso Cuarón        4
##  7 André Øvredal         3
##  8 Andrea Arnold         3
##  9 Andy Fickman          1
## 10 Andy Muschietti       3
## 11 Andy Tennant          1
## 12 Ang Lee               4
## 13 Angelina Jolie        2
## 14 Antoine Fuqua         2
## 15 Ava DuVernay          3
## 16 Barry Levinson        3
## 17 Barry Sonnenfeld      3
## 18 Baz Luhrmann          2
## 19 Ben Affleck           3
## 20 Ben Stiller           3
```

step3:display the top 10 H_index directors

```
dtf %>% top_n(10)
```

There are 17 directors with highest Hidden.Gem.Score H_INDEX = 4

```
## Selecting by H_index
```

```
## # A tibble: 17 x 2
##    directors            H_index
##    <chr>                <chr>
##  1 Alfonso Cuarón       4
##  2 Ang Lee              4
##  3 Bong Joon Ho         4
##  4 Christopher Nolan    4
##  5 David Fincher        4
##  6 David Mackenzie      4
##  7 Edgar Wright         4
```

```
##  8 Hayao Miyazaki       4
##  9 Martin Scorsese      4
## 10 Paul Thomas Anderson 4
## 11 Pedro Almodóvar      4
## 12 Peter Jackson        4
## 13 Quentin Tarantino    4
## 14 Ridley Scott         4
## 15 Steven Soderbergh    4
## 16 Steven Spielberg     4
## 17 Woody Allen          4
```