Final Project - MATH 208

Xiaoteng Zhang, Malek Hassouneh

11/21/2021

Task 1

Preliminary - Cleaning Data

In this segment, we first import the data (line 1) and do some cleaning, selecting movie data only (line 4) and dropping n.a values (line 7). We also keep track of the number of observations our tables have after applying each cleaning step (line 10). There is a substantial drop off in the number of observations, especially after removing any entries with NA values, though the number of observations is still large.

```
1  data= read.csv("Final_Project_FlixGem.csv")
2
3  # Selecting rows only containing Movie
4  movie= data%>% filter(grepl('Movie',Series.or.Movie))
5
6  # Rows with na are dropped
7  movie_no_na= movie%>%drop_na()
8
9  # Table to keep track of number of observations
10 tibble(Table = c("Data", " Data w/o Series", "Movie Data w/o NA"),
11        "Number of Observations"=c(nrow(data),nrow(movie),nrow(movie_no_na)))%>%
12   kable(caption = "Number of Observations Per Data Set")%>%
13   kable_classic(full_width = F, html_font = "Cambria", latex_options = "HOLD_position")
```

Table 1: Number of Observations Per Data Set

Table	Number of Observations
Data	9425
Data w/o Series	7010
Movie Data w/o NA	2118

a) Association between Hidden Gem Score, Run Time and Language Combinations

Here we are asked to check the association between hidden gem score and running time or language. First, some recoding of the variables (line 2), in which we take only the top 10 language combinations (more specifically languages with appearances above the 15 threshold), grouping the other language combinations under an "other" term (line 9):

```r
# Data.frame for top 10 language combinations appearing in the data set
language=movie_no_na%>%group_by(Languages)%>%tally()%>%arrange(desc(n))

head(language,10) %>%
  kable(caption = "Top 10 Language Combinations")%>%
  kable_classic(full_width = F, html_font = "Cambria",
                latex_options = "HOLD_position")
```

Table 2: Top 10 Language Combinations

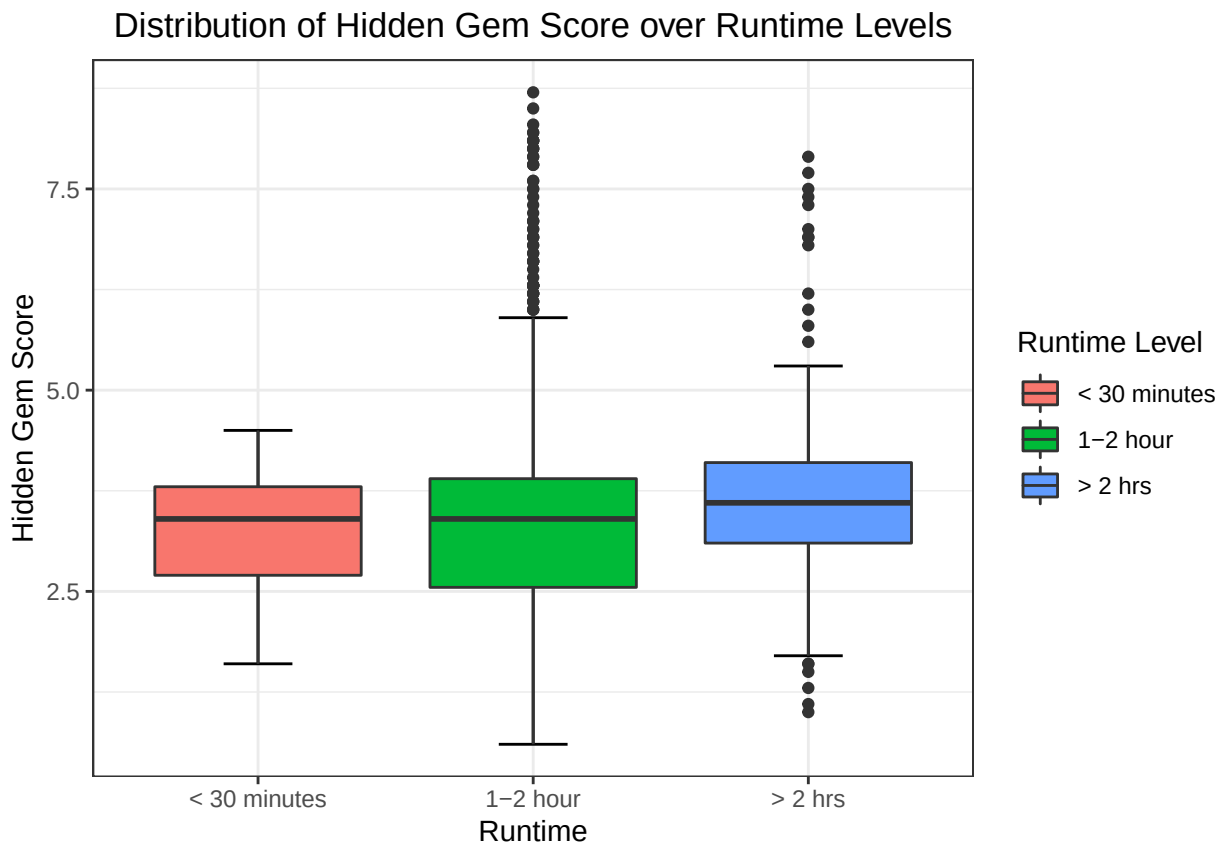
Languages	n
English	1025
English, Spanish	123
English, French	74
English, German	29
English, Italian	28
English, Russian	27
Japanese	21
English, Japanese	19
English, Ukrainian	17
English, Mandarin	15

```r
movie_top10= movie_no_na%>%mutate(language_lmp= fct_lump(movie_no_na$Languages%>%
                                                         as.factor,10))
```

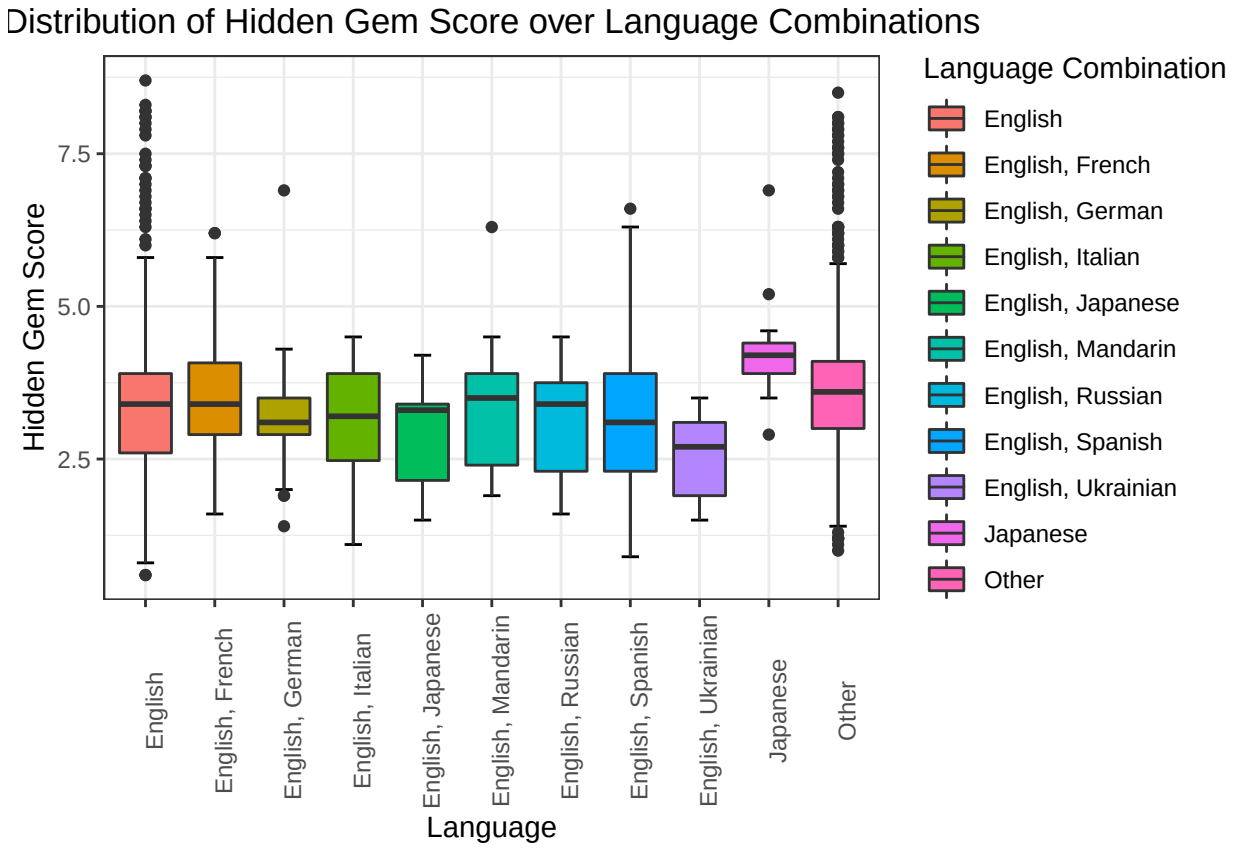
Now, we plot boxplots of the hidden gem scores over running time (line 6) and language combinations (line 1 of next chunk). Examining the first boxplot, there seems to be no discernible association between hidden gem scores and running time. This is seen by the relatively stable level of the median across the runtime levels, and the relatively narrow IQR for each box. Comparatively, there is slightly more variation in hidden gem scores across language combinations, though the association is still negligible. This is distinctly seen by looking at the median values of the boxes (for instance, Japanese having a larger median value than the rest), though the IQR of the boxes in the language plot are also significantly wider compared to the runtime plot (first boxplot).

```r
## Reordering factor levels of Runtime for the purpose of graphing
movie_top10 = movie_top10%>%
  mutate(Runtime_new=fct_relevel(Runtime,c("< 30 minutes","1-2 hour","> 2 hrs")))

#Association between hidden gem score and running time
ggplot(movie_top10,aes(x=Runtime_new,y=Hidden.Gem.Score,fill=Runtime_new))+
  stat_boxplot(geom="errorbar",width=0.25)+ geom_boxplot() + theme_bw() +
  labs(title="Distribution of Hidden Gem Score over Runtime Levels", y="Hidden Gem Score",
       x="Runtime", fill = "Runtime Level") + theme(plot.title =
                                                      element_text(hjust = 0.5))
```




```
#Association between hidden gem score and language
ggplot(movie_top10,aes(x=language_lmp,y=Hidden.Gem.Score,fill=language_lmp))+
  stat_boxplot(geom="errorbar",width=0.25)+ geom_boxplot() + theme_bw() +
  labs(title="Distribution of Hidden Gem Score over Language Combinations",
       y="Hidden Gem Score", x="Language", fill = "Language Combination") +
  theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 90))
```

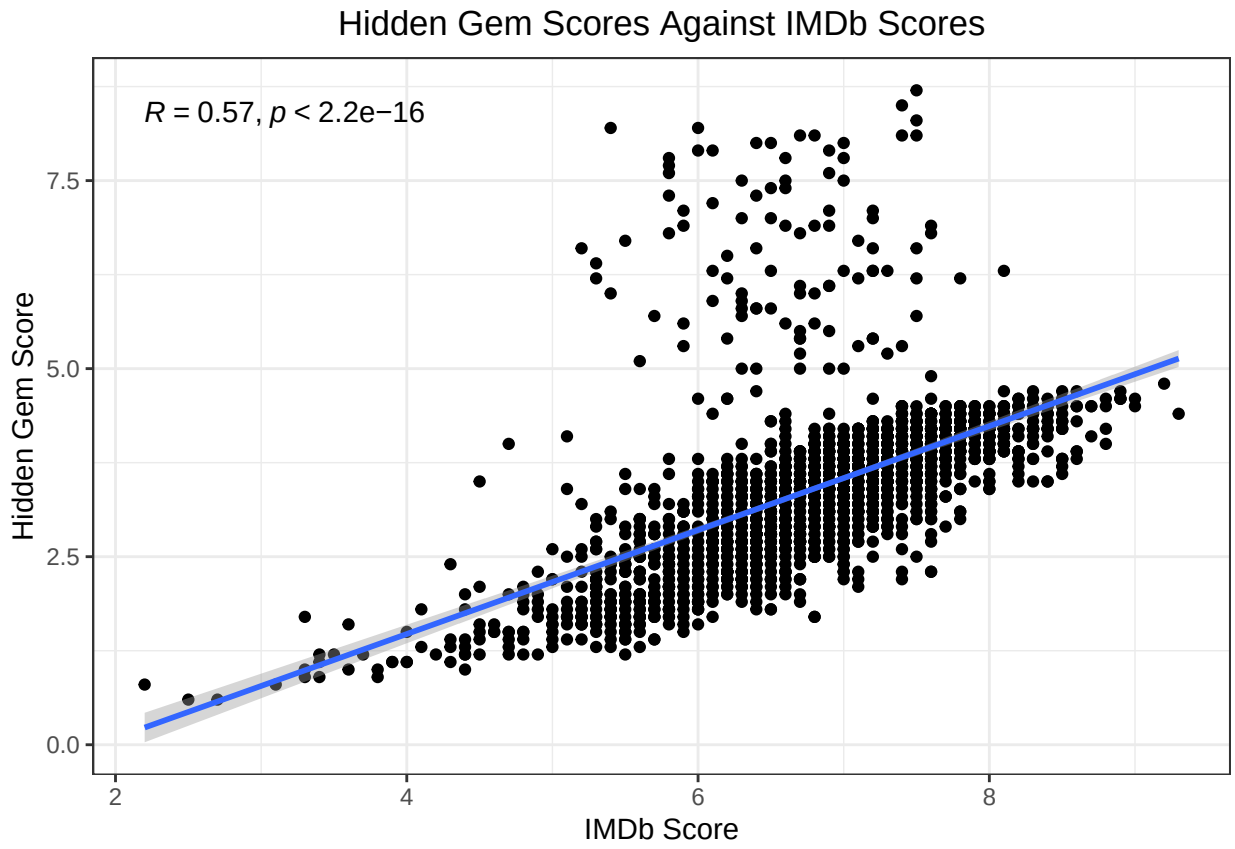


b) Association between Hidden Gem Score and Review Site Scores

To test the association between hidden gem score and different review site scores, we plot the pairs of scores for each movie (lines 2 of each chunk) in a scatter plot, plotting a line of best fit to aid with visualization. The sample correlation between each of the pairs of variables is also included in the top left. There seems to be a strong positive association between hidden gem scores and Rotten Tomatoes and Metacritic, with sample correlations of 0.79 and 0.74 respectively. Comparatively, the association with IMDb scores is also positive, though weaker, with a sample correlation of only 0.57.

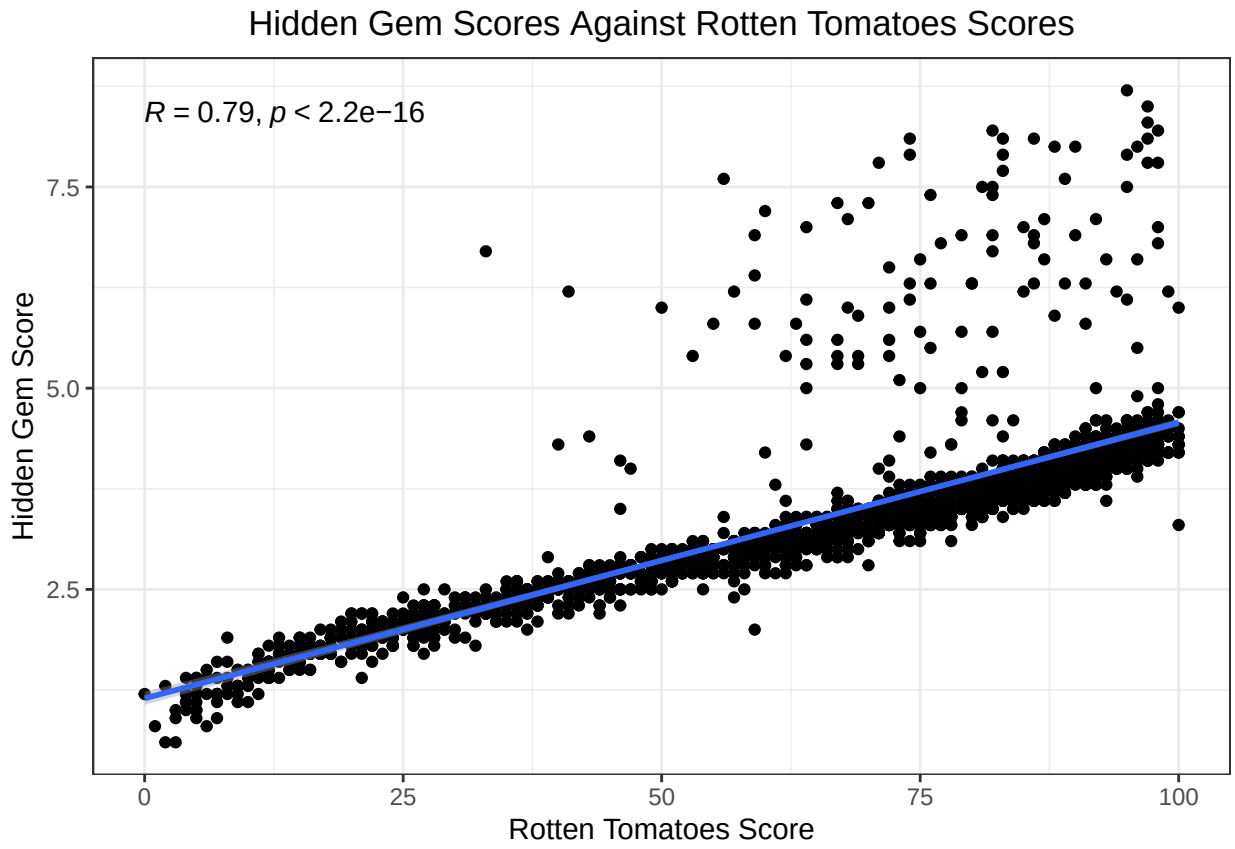
```
1 #IMDb Score
2 ggplot(movie_top10,aes(x=IMDb.Score,y=Hidden.Gem.Score))+ geom_point()+ theme_bw() +
3   labs(title = "Hidden Gem Scores Against IMDb Scores", x="IMDb Score",
4        y="Hidden Gem Score")+ geom_smooth(method='lm')+ stat_cor() +
5   theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



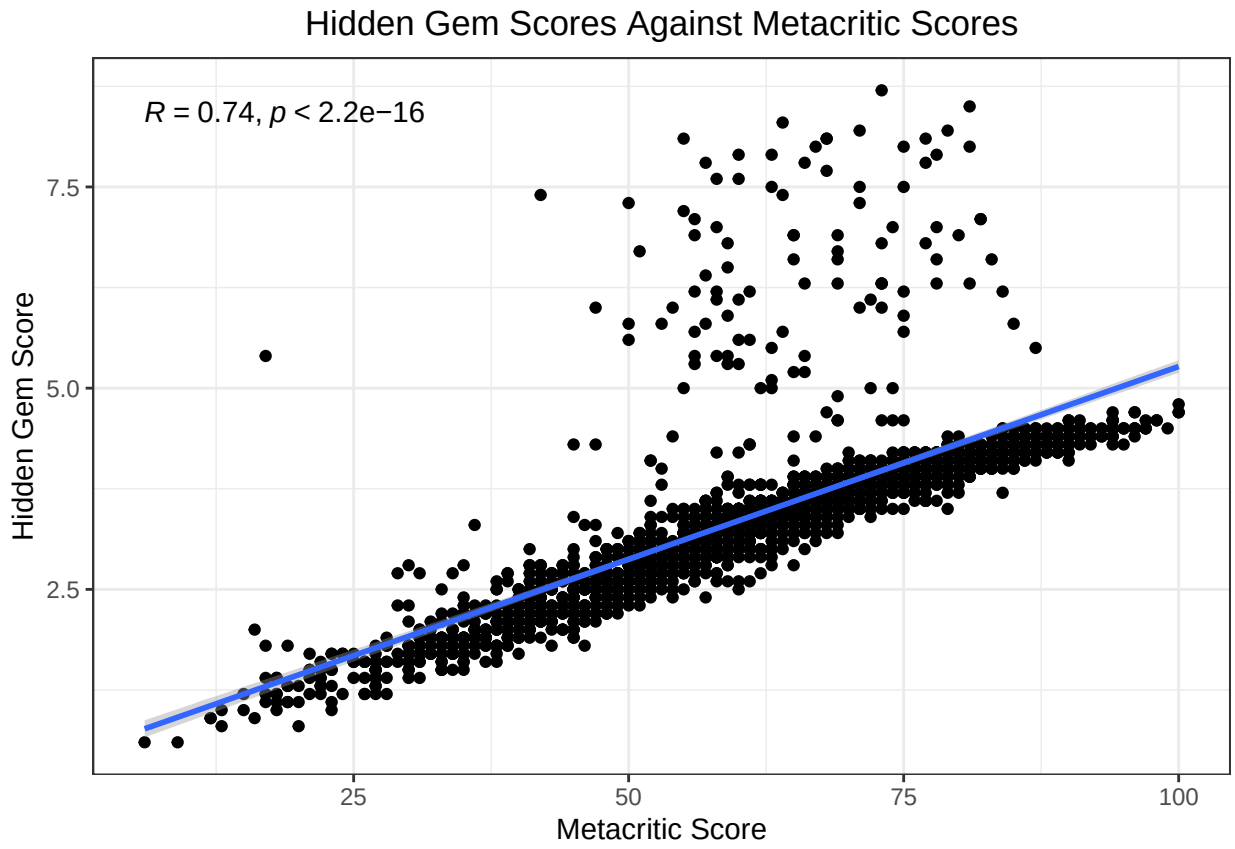

```
#Rotten Tomato
ggplot(movie_top10,aes(x=Rotten.Tomatoes.Score,y=Hidden.Gem.Score))+ geom_point() +
  theme_bw() +
  labs(title = "Hidden Gem Scores Against Rotten Tomatoes Scores",
       x="Rotten Tomatoes Score", y="Hidden Gem Score")+ geom_smooth(method='lm')+
  stat_cor() + theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
#Metacritic Score
ggplot(movie_top10,aes(x=Metacritic.Score,y=Hidden.Gem.Score))+ geom_point() +
  theme_bw() +
  labs(title = "Hidden Gem Scores Against Metacritic Scores", x="Metacritic Score",
       y="Hidden Gem Score")+ geom_smooth(method='lm')+ stat_cor() +
  theme(plot.title = element_text(hjust = 0.5))
```

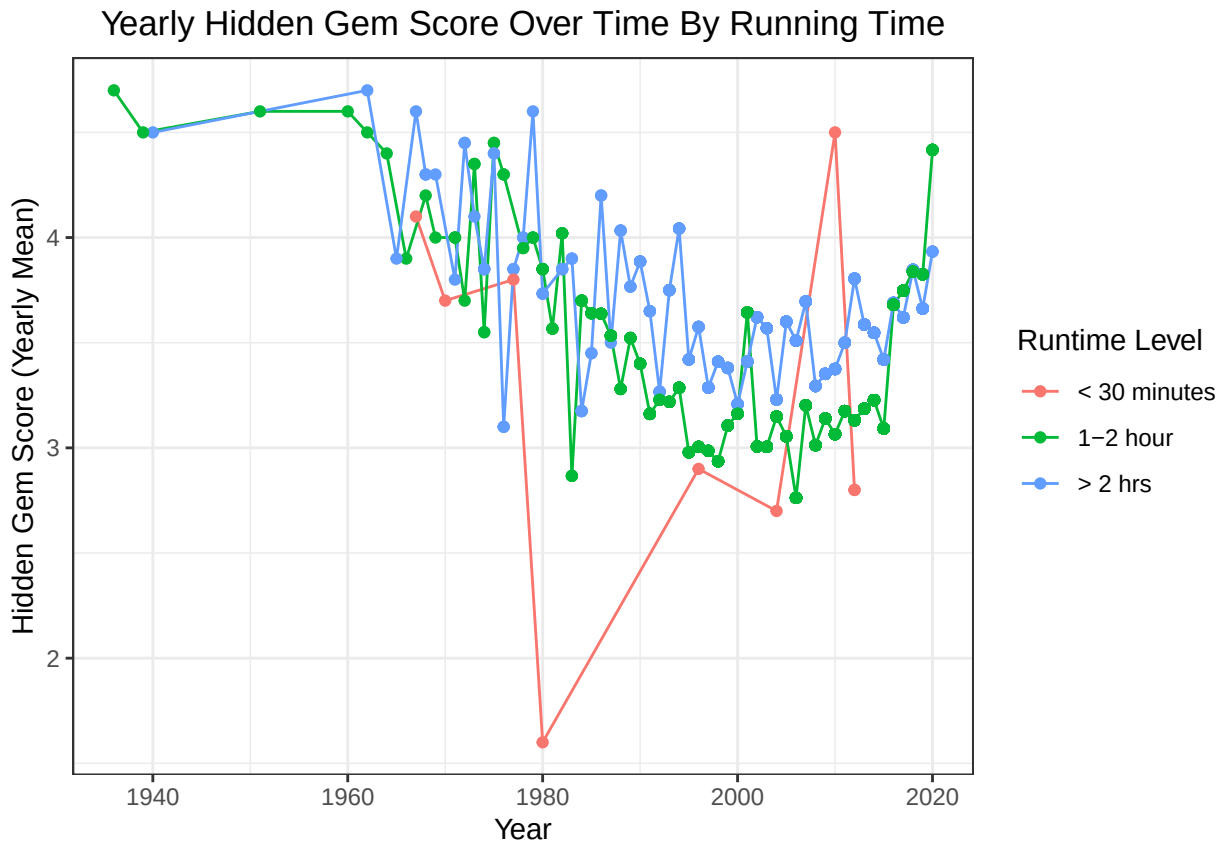
```
## `geom_smooth()` using formula 'y ~ x'
```



c) Hidden Gem Scores Over Time by Runtime Length

To smooth out the trends in the graph, yearly averages of the hidden gem scores were taken for each runtime category (line 5). The graph (line 8) shows that hidden gem scores of movies with runtimes of "1-2 hour" and ">2 hrs" are trending upwards as of 2010, though the increase in the score of "1-2 hour" movies is larger. Hence, it is not the case that longer movies' ratings are improving strictly due to Netflix, as the score of shorter movies is also improving in that time, and by a greater amount.

```
1  #Date reformating
2  movie_top10$Release.Date=as.Date(movie_top10$Release.Date)
3
4  #Grouping and averaging yearly
5  movie_group=movie_top10%>%mutate(Year = floor_date(Release.Date,unit="year"))%>%
6    group_by(Year,Runtime_new)%>%mutate(Yearly_mean=mean(Hidden.Gem.Score))
7
8  ggplot(movie_group,aes(x=Year,y=Yearly_mean, group=Runtime_new,col=Runtime_new))+
9    geom_line() + geom_point() + theme_bw()  +
10   labs(title = "Yearly Hidden Gem Score Over Time By Running Time", x="Year",
11        y="Hidden Gem Score (Yearly Mean)")+
12   theme(plot.title = element_text(hjust = 0.5))+ scale_colour_discrete("Runtime Level")
```



Task 2

The first 3 nodes of the regression tree partition the data based on Rotten Tomatoes Score, clearly showing that it is the most relevant feature when predicting hidden gem scores. IMDb Score is the second most important feature, increasing the predictive performance of the model only when Rotten Tomatoes Score > 94.5. To assess the predictive performance of the model holistically, we can look at the decrease in the deviance (the second to last number in each line of the model output) prior to introducing a new node/partition. What can be seen is that the deviance experiences a large decrease if Rotten Tomatoes Score < 32.5, i.e. the model has high predictive power if Rotten Tomatoes score is relatively low. Comparatively, if Rotten Tomatoes scores are high, > 94.5, and IMDb Scores are less than 6.75, then the model also has high predictive power, decreasing the deviance the most compared to its original value. Anything in between, i.e. $58.5 <$ Rotten Tomatoes Score < 94.5, the model exhibits relatively lower predictive power. As a list, the model is most predictive for movies given:

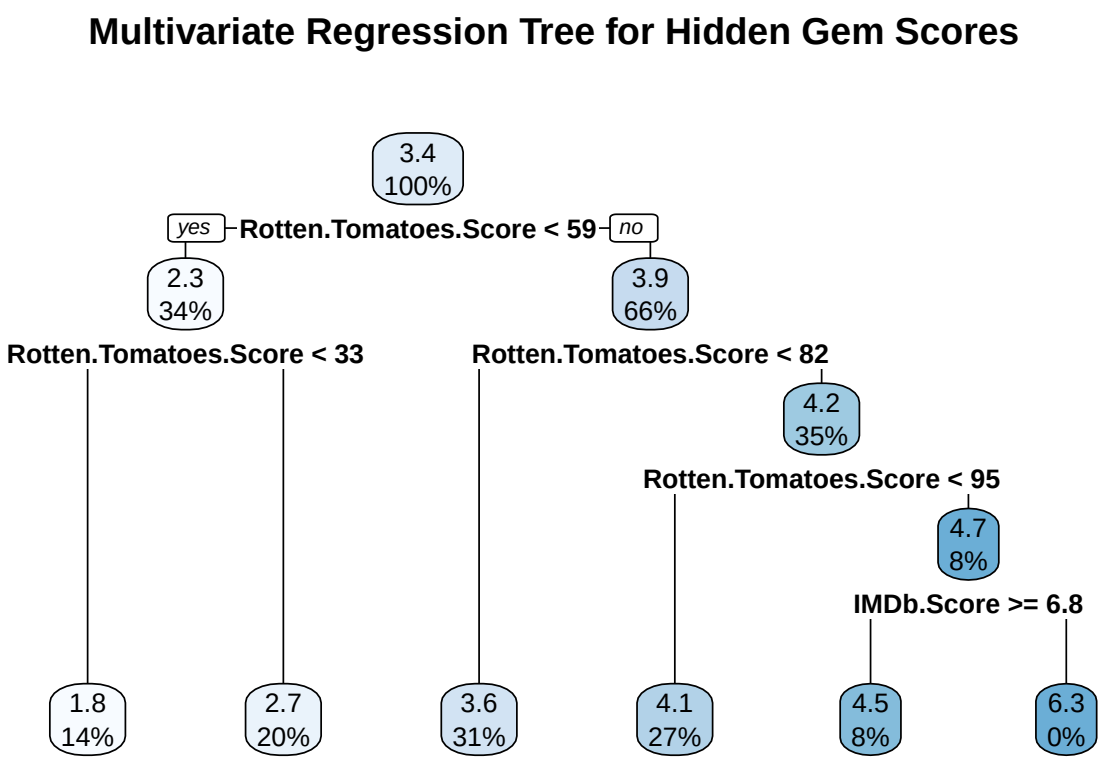
1) Rotten Tomatoes Score > 94.5 & IMDb Score < 6.75
2) Rotten Tomatoes Score < 32.5
3) Rotten Tomatoes Score > 94.5 & IMDb Score > 6.75
4) $32.5 <$ Rotten Tomatoes Score < 58.5
5) $58.5 <$ Rotten Tomatoes Score < 94.5 (rest of divisions)

```
#Creating the Regression Tree Model
m1= rpart(formula = Hidden.Gem.Score~
          Runtime+language_lmp+IMDb.Score+Metacritic.Score+Rotten.Tomatoes.Score,
```

```
          data=movie_top10,method = "anova")
m1
```

```
## n= 2118
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 2118 2548.05300 3.396412
##    2) Rotten.Tomatoes.Score< 58.5 716  314.62640 2.347765
##      4) Rotten.Tomatoes.Score< 32.5 299   40.34562 1.809030 *
##      5) Rotten.Tomatoes.Score>=32.5 417  125.27650 2.734053 *
##    3) Rotten.Tomatoes.Score>=58.5 1402 1043.96800 3.931954
##      6) Rotten.Tomatoes.Score< 81.5 664  407.27110 3.583133 *
##      7) Rotten.Tomatoes.Score>=81.5 738  483.21200 4.245799
##       14) Rotten.Tomatoes.Score< 94.5 568  287.35490 4.124648 *
##       15) Rotten.Tomatoes.Score>=94.5 170  159.66490 4.650588
##         30) IMDb.Score>=6.75 160  101.79940 4.548125 *
##         31) IMDb.Score< 6.75 10   29.30900 6.290000 *
```

```
#Plotting the Regression Tree
rpart.plot(m1, main="Multivariate Regression Tree for Hidden Gem Scores")
```



Task 3

As a first observation, the HG-H index must be an integer, as it is simply a discrete count variable. Calculating this index can be done as follows: 1) finding the number of movies with a hidden gem score greater than "h" 2) finding the max "h" under which the calculated sum from step 1 is greater than the value of "h", which will then be the HG-H index. This is demonstrated in the function below (starting line 1) . A tibble displaying the directors with the highest HG-H scores is also included below.

```
h_index<-function(x){

  #Creating an empty vector to store outputs of the loop below
  a<-rep(NA,round(max(x),2))

    for(h in 1:length(a)){
    #Finding the number of movies that have a hidden gem score greater than "h"
      a[h]<-sum(x>=h)
    }

    #A technical step which assigns an HG-H score of 0 if a director has no movies
    #with a hidden gem score greater than "h" for all "h"
    if(max(a)==0){
      h_score=0
    }

    #Defining the HG-H index as the max index for which the number of movies
    #with a hidden gem score greater than "h" is greater than "h"
    else{
      h_score<-max(which((a>=1:length(a))))
    }
      h_score
}

movie_final=movie_no_na[,c(9,6)]%>%group_by(Director)%>%summarise(
  HG_H_Score = h_index(Hidden.Gem.Score))%>%arrange(desc(HG_H_Score))

movie_final%>%head(15)%>%
  kable(caption = "Top 15 Directors based on HG-H Score (Preliminary)")%>%
  kable_classic(full_width = F, html_font = "Cambria",
                latex_options = "HOLD_position")
```

Table 3: Top 15 Directors based on HG-H Score (Preliminary)

Director	HG_H_Score
Alfonso CuarÃ³n	4
Ang Lee	4
Bong Joon Ho	4
Christopher Nolan	4
David Fincher	4
David Mackenzie	4
Edgar Wright	4
Hayao Miyazaki	4
Martin Scorsese	4
Paul Thomas Anderson	4
Pedro AlmodÃ³var	4
Peter Jackson	4
Quentin Tarantino	4
Ridley Scott	4
Steven Soderbergh	4

There is clearly more than 10 directors with HG-H scores of 4, hence a second criterion to differentiate between them and find the top 10 is needed, say, the number of movies directed by each director (in this data set). A final tibble (line 15) with the directors with the highest HG-H scores ranked by the number of movies they have directed is included below:

```
1  #More than 10 directors have an HG-H Score greater than 4, hence another criterion
2  #other than HG-H score must be used to differentiate them
3  top_directors=filter(movie_final,HG_H_Score>=4)
4
5  #Counting the number of appearances (movies) for each of the top directors
6
7  a= NULL
8
9  for (i in seq_along(top_directors$Director)) {
10   a[i]= sum(str_count(movie_no_na, toString(top_directors[i,1])))
11 }
12
13 #Final tibble with top 10 directors, all with HG-H score = 4
14
15 cbind(top_directors,a)%>%arrange(desc(a))%>%rename(Appearances=a)%>%head(10)%>%
16   kable(caption = "Top 10 Directors based on HG-H Score (Ranked by Number of Movies)")%>%
17   kable_classic(full_width = F, html_font = "Cambria",
18                 latex_options = "HOLD_position")
```

Table 4: Top 10 Directors based on HG-H Score (Ranked by Number of Movies)

Director	HG_H_Score	Appearances
Steven Spielberg	4	32
Woody Allen	4	27
Pedro AlmodÃ³var	4	26
Quentin Tarantino	4	18
Christopher Nolan	4	17
Hayao Miyazaki	4	16
Martin Scorsese	4	15
Peter Jackson	4	15
Ridley Scott	4	13
Steven Soderbergh	4	13