Math208 Final Project

Joseph Wu, Darren Bugaresti, Anna Hayes

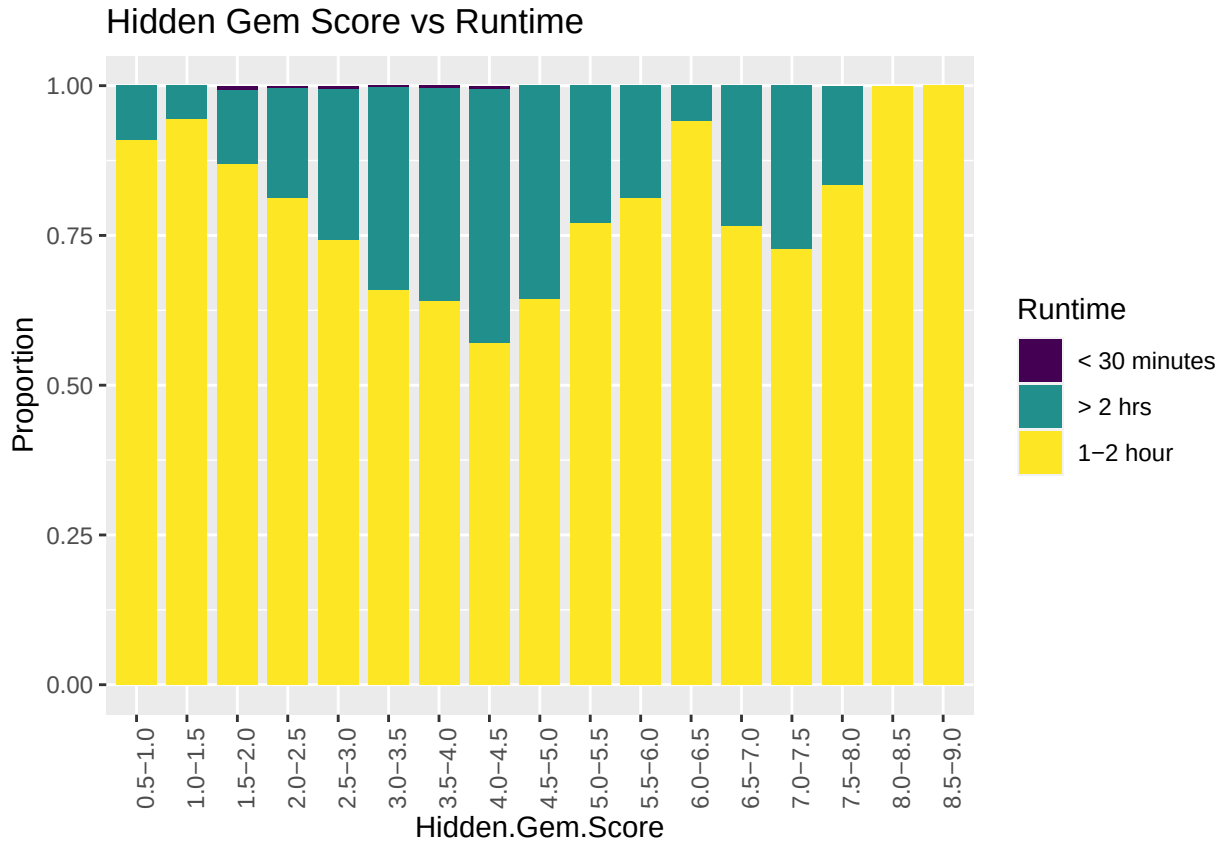
12/1/2021

```
#drop na and select only movies
project<-read.csv("Final_Project_FlixGem.csv")%>%drop_na%>%as_tibble%>%filter(Series.or.Movie=="Movie")
```

Task 1 : Data Wrangling and Exploratory Data Analyses

a. Does the Hidden Gem Score seems to be associated to the Runtime Category or the languages used in the film? Explain briefly the reasons behind your assessment. Hint: You may need to do some re-coding of one or both of these variables. Any reasonable re-coding is fine, just be sure to be clear what you've done.

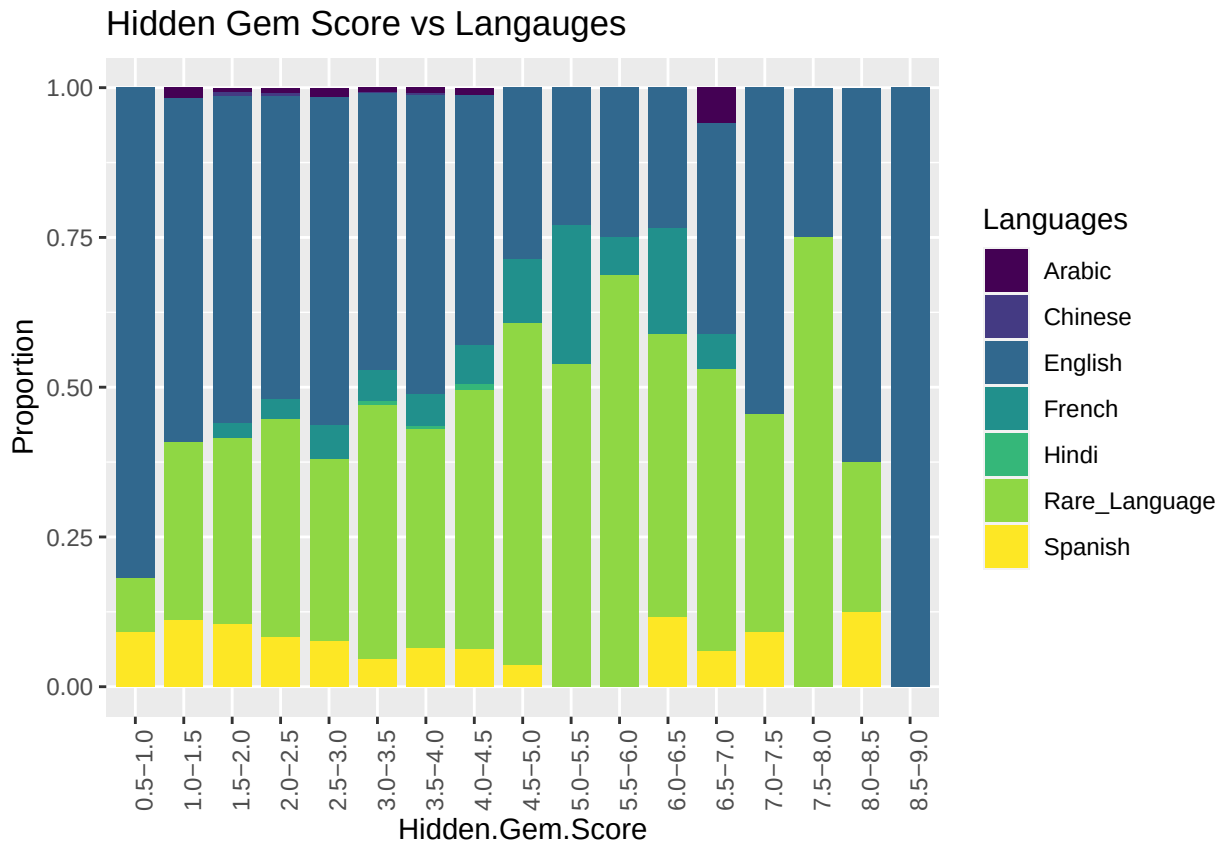
```
#group Hidden.Gem.score into 17 subgroups
lbls <-
  c( '0.5-1.0', '1.0-1.5',  '1.5-2.0',  '2.0-2.5', '2.5-3.0',
     '3.0-3.5', '3.5-4.0',  '4.0-4.5',  '4.5-5.0', '5.0-5.5',
     '5.5-6.0', '6.0-6.5',  '6.5-7.0',  '7.0-7.5', '7.5-8.0',
     '8.0-8.5', '8.5-9.0')
breaks = c(0.5 ,1.0, 1.5 ,2.0 ,2.5 ,3.0 ,3.5 ,4.0 ,
           4.5 ,5.0 ,5.5 ,6.0 ,6.5 ,7.0 ,7.5 ,8.0 ,
           8.5 ,9.0)
data <-project %>% mutate(HGS_group = cut(
             Hidden.Gem.Score,
             breaks = breaks,
             right = T,
             labels = lbls ))

data<-data%>%ungroup()%>%group_by(HGS_group)%>%mutate(Count=n())
data<-data%>%group_by(HGS_group) %>% mutate(Proportion =Count / sum(Count))


ggplot(data, aes(x = HGS_group, y= Proportion, fill = Runtime, width = .8)) +
  geom_bar(stat = "identity") +scale_fill_viridis_d() +
  xlab("Hidden.Gem.Score")  +labs(title = "Hidden Gem Score vs Runtime")+
  theme(axis.text.x = element_text(angle = 90))
```


Hidden Gem Score vs Runtime
1.00
0.75
0.50
0.25
0.00
Proportion
0.5−1.0
1.0−1.5
1.5−2.0
2.0−2.5
2.5−3.0
3.0−3.5
3.5−4.0
4.0−4.5
4.5−5.0
5.0−5.5
5.5−6.0
6.0−6.5
6.5−7.0
7.0−7.5
7.5−8.0
8.0−8.5
8.5−9.0
Hidden.Gem.Score
Runtime
< 30 minutes
> 2 hrs
1−2 hour

```r
#too many languages, manipulation with the languages here..
temp <- drop_na(project)[c("Languages")] %>% unique()
input_fun <-
  function(x) {#helper function 1: split the string
    str_split(x, ", ")
  }
unique_language <-
  lapply(temp, input_fun) %>% unlist() %>% unique()
length(unique_language)
```

```
## [1] 121
```

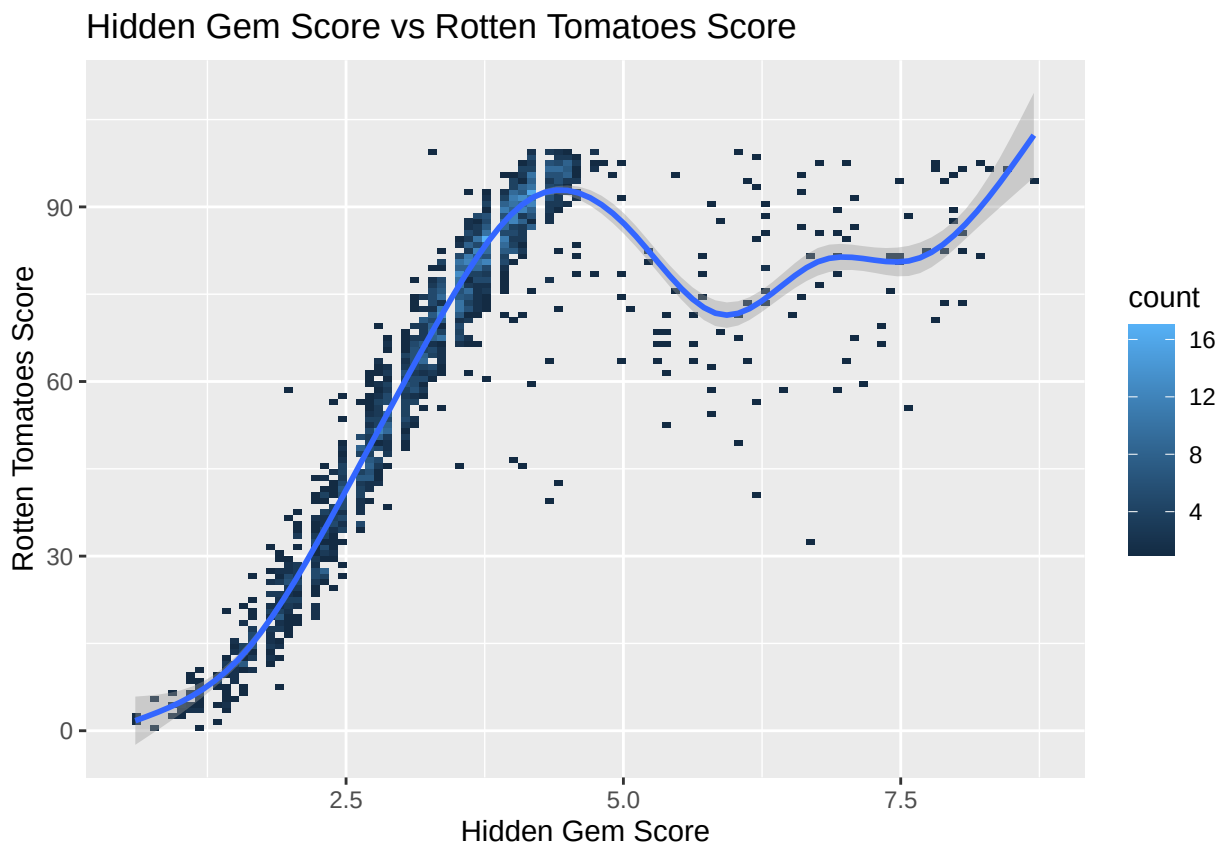
```r
#there are total 121 unique languages, now we group the languages
#select top six most spoken languages into 6 different groups and rest languages into a single group
Lgroup1 <- c("English")
Lgroup2 <- c("Spanish")
Lgroup3 <- c("French")
Lgroup4 <- c("Hindi")
Lgroup5 <- c("Chinese")
Lgroup6 <- c("Arabic")
Lgroup7 <-
 unique_language[!unique_language %in% c("English", "Spanish", "French", "Hindi", "Chinese", "Arabic")]
input_fun2 <-
  function(x, g1, g2, g3, g4, g5, g6, g7) {
    # helper function 2: x is the original language group and return the new language group!
    TF_array7 <- input_fun(x) %>% unlist() %in% g7
    TF_array6 <- input_fun(x) %>% unlist() %in% g6
    TF_array5 <- input_fun(x) %>% unlist() %in% g5
    TF_array4 <- input_fun(x) %>% unlist() %in% g4
    TF_array3 <- input_fun(x) %>% unlist() %in% g3
    TF_array2 <- input_fun(x) %>% unlist() %in% g2
    TF_array1 <- input_fun(x) %>% unlist() %in% g1
    if (TRUE %in% TF_array7) { return("Rare_Language") }
    if (TRUE %in% TF_array6) { return("Arabic")}
    if (TRUE %in% TF_array5) { return("Chinese")}
    if (TRUE %in% TF_array4) { return("Hindi")}
    if (TRUE %in% TF_array3) { return("French")}
    if (TRUE %in% TF_array2) { return("Spanish") }
    if (TRUE %in% TF_array1) { return("English") }
  }
for (i in 1:length(data$Languages)) {
  #loop around the char vector and change every single cell!
 data$Languages[i] <-
    input_fun2(
    data$Languages[i], Lgroup1, Lgroup2,Lgroup3, Lgroup4, Lgroup5,
      Lgroup6, Lgroup7
    )
}
#languages manipulation ends here, now plot!
ggplot(data, aes(x = HGS_group, y= Proportion, fill = Languages, width = .8)) +
  geom_bar(stat = "identity") +scale_fill_viridis_d() +
  xlab("Hidden.Gem.Score") + labs(title = "Hidden Gem Score vs Langauges")+
  theme(axis.text.x = element_text(angle = 90))
```



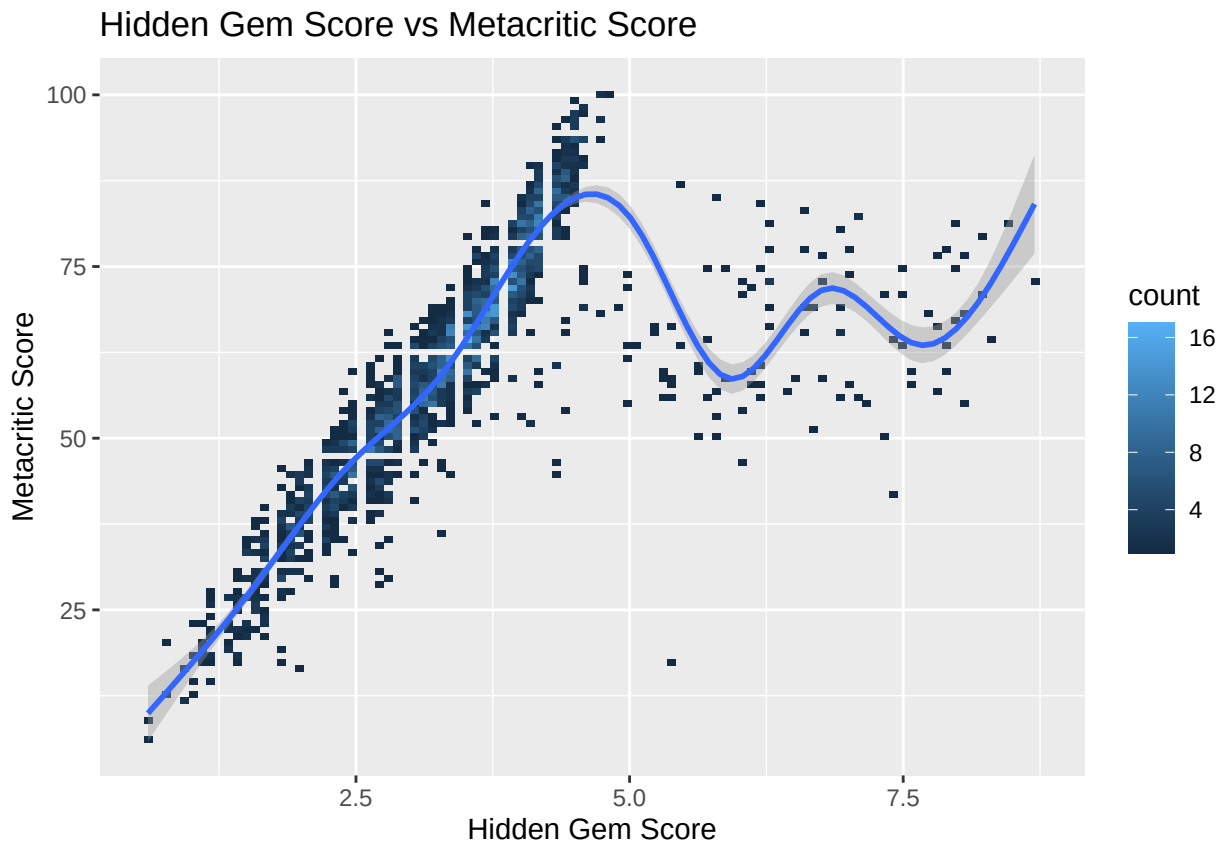
The Hidden Gem Score seems to be most associated to the Runtime, since the proportions fluctuate more with the levels of the runtime category.
With languages, the proportions of languages for each Hidden Gem Score interval are about the same. Meanwhile, with Runtime, we can see how the proportions of runtime vary with the hidden gem score, with films over 2 hours increasing in Hidden Gem Score.

b. Do any of the three review site scores (IMDb, Rotten Tomatoes, Metacritic) seem to be strongly or weakly correlated with the Hidden Gem Scores? Explain briefly the reasons behind your assessment and the nature of those associations.

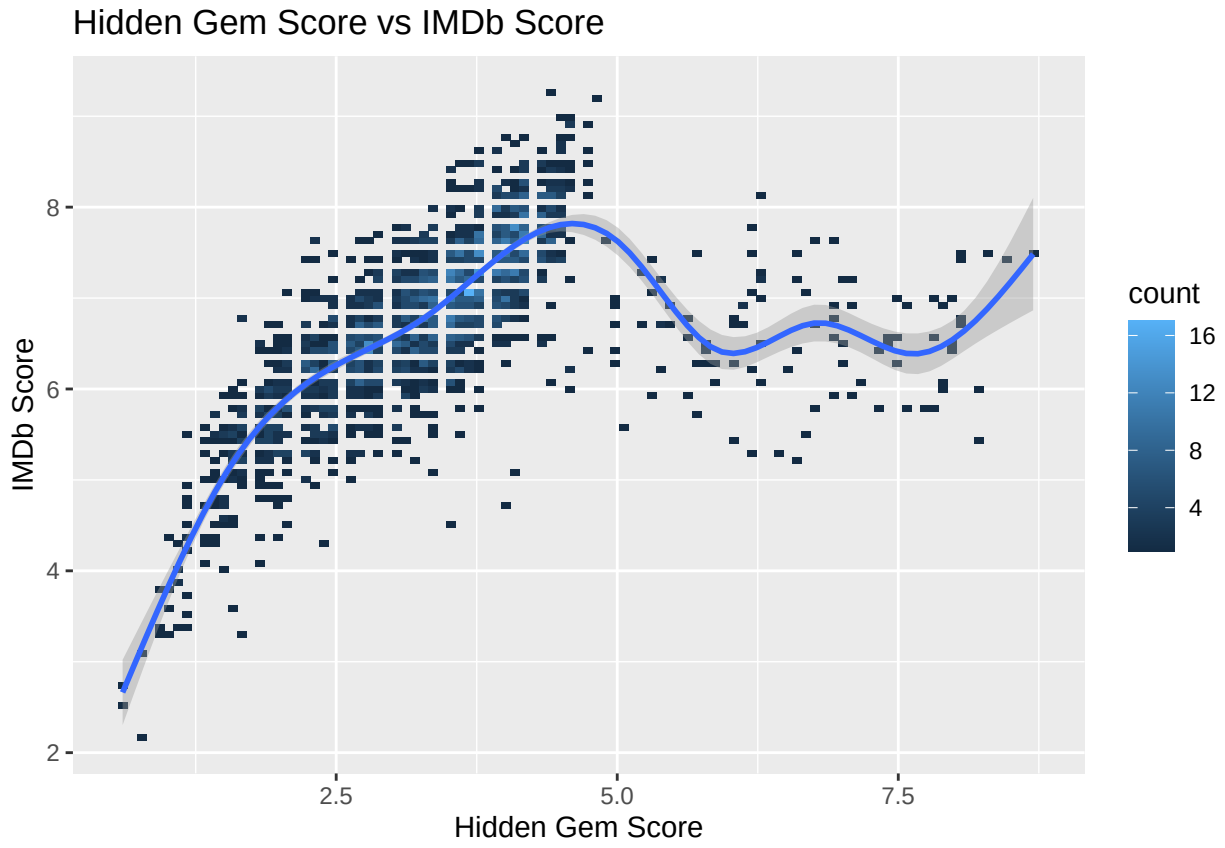
```
ggplot(project,aes(x=Hidden.Gem.Score,y=Rotten.Tomatoes.Score))+geom_bin2d(bins=100)+geom_smooth() +
  labs(title = "Hidden Gem Score vs Rotten Tomatoes Score",
       x = "Hidden Gem Score", y = "Rotten Tomatoes Score")
```



```
ggplot(project,aes(x=Hidden.Gem.Score,y=Metacritic.Score))+geom_bin2d(bins=100)+geom_smooth()+
  labs(title = "Hidden Gem Score vs Metacritic Score",
       x = "Hidden Gem Score", y = "Metacritic Score")
```



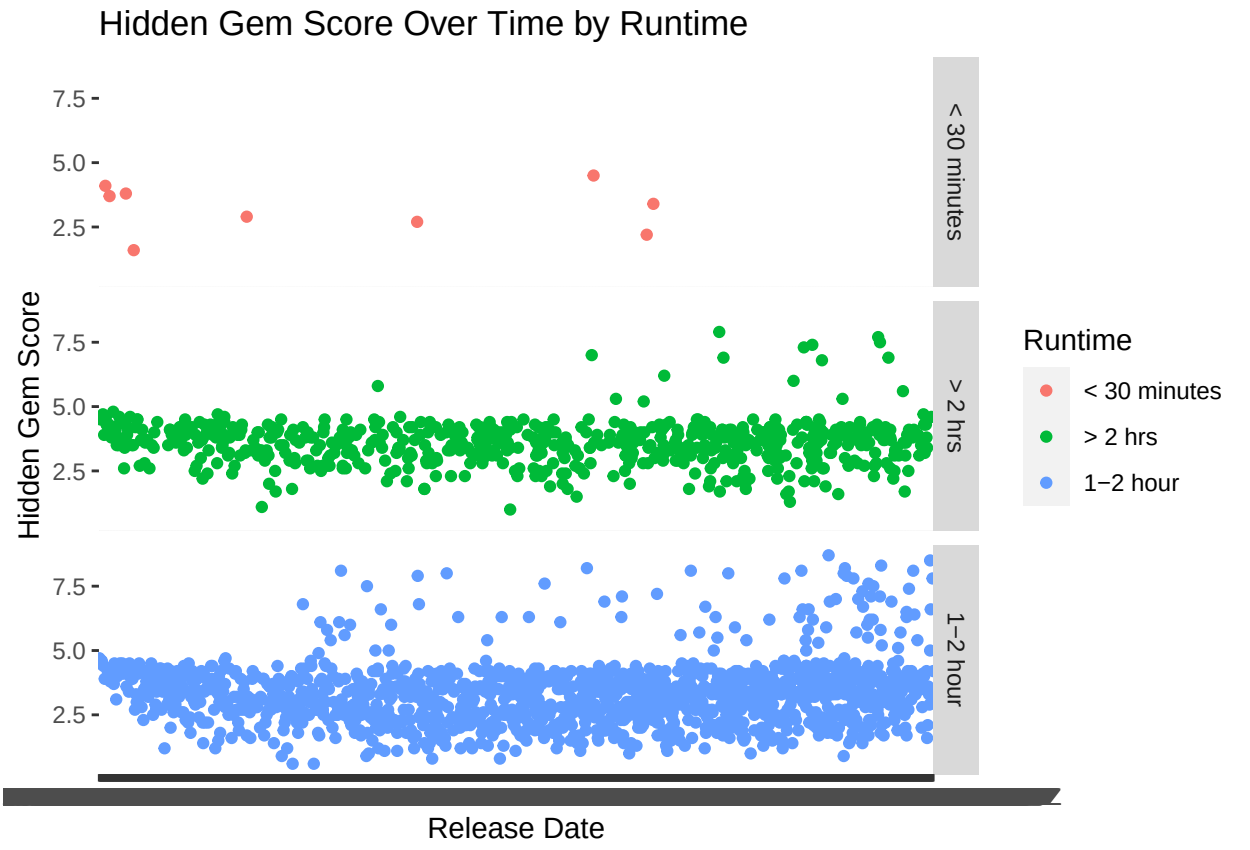
```
ggplot(project,aes(x=Hidden.Gem.Score,y=IMDb.Score))+geom_bin2d(bins=100)+geom_smooth() +
  labs(title = "Hidden Gem Score vs IMDb Score",
       x = "Hidden Gem Score", y = "IMDb Score")
```



Based on the smoothing lines, we can see trends in the data relating the Hidden Gem Score to other scores and can conclude that the Rotten Tomatoes Score is strongly related to Hidden Gem Score as most of the data points are closely compact to the line, which has a steep slope. The line showing the relationship between the IMDB Score and the Hidden Gem Score curves more, the slope is not as steep towards the middle, and the data points are further spread out, meaning the two are more weakly correlated from the above three graphs. However, for all three graphs, the scores are not strongly associated with high Hidden Gem Scores (scores >5.0).

c. The company has a theory that people are becoming more acceptable of longer movies because they can watch them at home on Netflix and other content-collecting sites. Do you notice any trend over time in the Hidden Gem Scores by category of RunTime Length? Explain briefly the reasons behind your assessment.

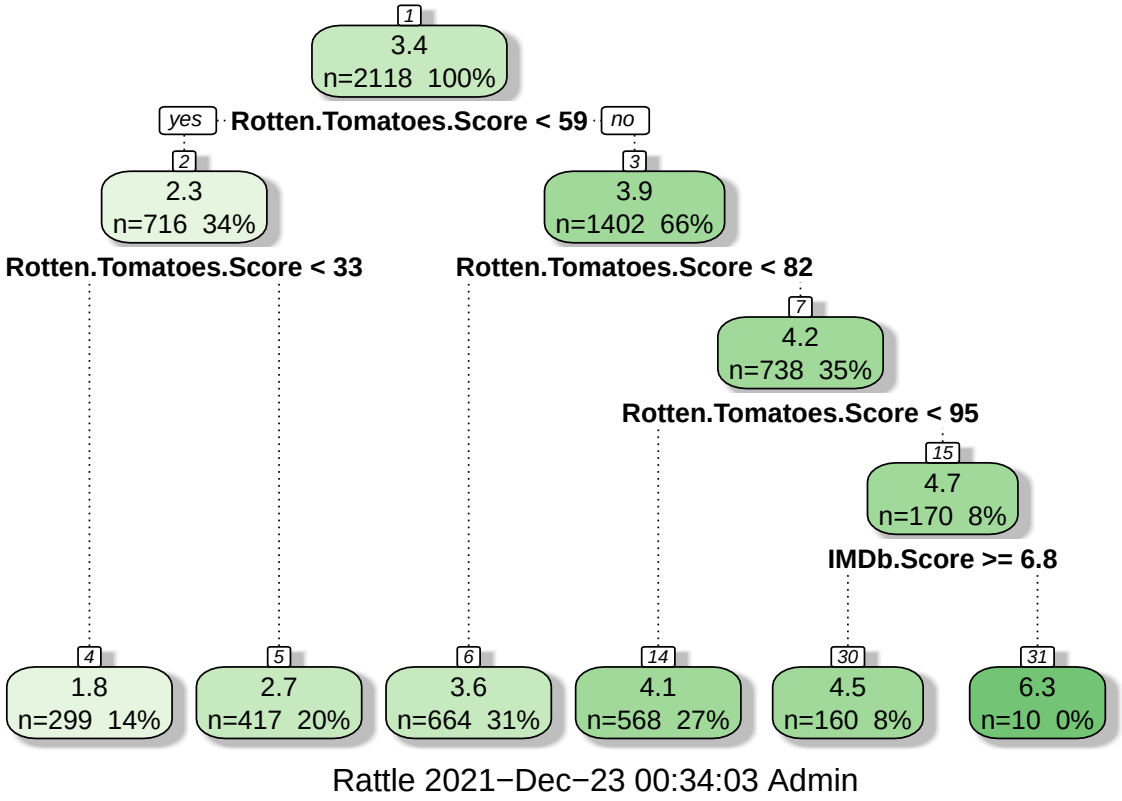
```
# x-axis: Release Date,
# y-axis: HGS,
# three graphs corresponding to the different Runtime

ggplot(project, aes(x = Release.Date, y = Hidden.Gem.Score)) +
  geom_point(aes(color = Runtime)) + facet_grid('Runtime') +
  labs(title = "Hidden Gem Score Over Time by Runtime",
       x = "Release Date", y = "Hidden Gem Score")
```



Within the last decade, longer movies (with runtimes > two hours) are being given higher Hidden Gem Scores. On the scatterplot, it may be observed that before the year 2000, the highest hidden gem score that movies over 2 hours aschieved was about a 5.0, whereas over a dozen movies have been plotted at over the 5.0 mark since 2000. However, this could be associated with the fact that there are more longer movies being made over time, as the number of points plotted increases in that same time frame. Furthermore, 1-2 hour long movies have also increased in number since 2000, as seen on the graph and also have consistently been placed at over a 5.0 hidden gem score after not receiving that high of a score before 2000.

Task2 : Regression tree

```
#select predictors

task2_df<-data%>%ungroup()%>%
 select("Languages","Runtime","Metacritic.Score","IMDb.Score",
            "Rotten.Tomatoes.Score","Hidden.Gem.Score")

reg_tree<-rpart(Hidden.Gem.Score ~.,
                data=task2_df,
                method="anova"  )

fancyRpartPlot(reg_tree)
```




```
pred <- predict(reg_tree, newdata = data)
RMSE(pred = pred, obs = data$Hidden.Gem.Score)
```

```
## [1] 0.684151
```

Based on the regression tree, the most important feature for predicting the Hidden Gem Score appears to be the Rotten Tomatoes Score. This is because Rotten.Tomatoes have the same class as response vector (Hidden.Gem.Score) and rpart function make intelligent guesses for splitting the data.
The final RMSE is 0.684151 which suggests that, on average, our predicted Hidden.Gem.Scores are about 0.684151 off from the actual Hidden.Gem.Score.