TASK 1

```
4  mydata <- read.csv("Final_Project_FlixGem.csv")
5
6
7  mydata <- mydata %>% select(Title, Languages, Series.or.Movie, Hidden.Gem.Score, Runtime, Director, IMDb.Score, Rotten.Tomatoes.Score,
8                              Metacritic.Score, Release.Date, Summary)
9
10
11 mydata <- mydata %>% filter(Series.or.Movie == 'Movie')
12
13 nrow(mydata)
14 ncol(mydata)
15
16 mydata <- na.omit(mydata)
17
18 nrow(mydata)
19 ncol(mydata)
20
```

Line 4 imports the csv file and stores the data in a variable called mydata. Line 7, 8 selects only the rows mentioned in the pdf document from the dataset, while line 16 gets rid of all missing variable values from the dataset. Finally, after all the cleanup, the dataset mydata is left with 3661 rows and 11 columns.

Part a.

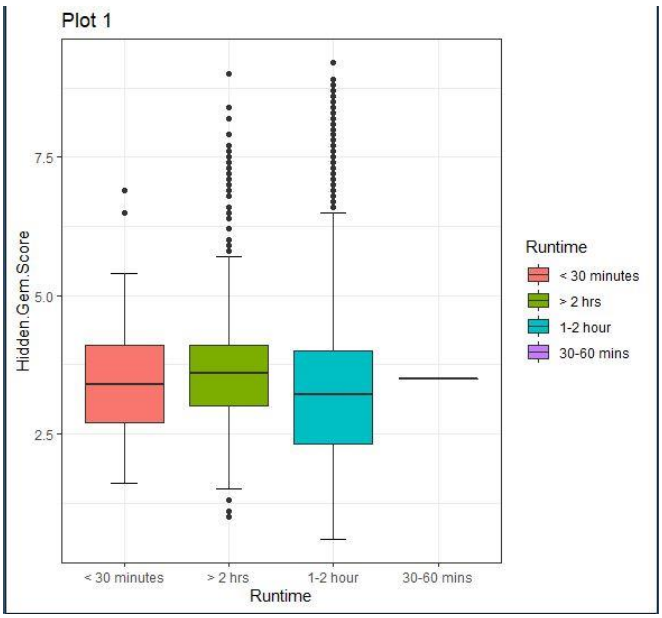


Figure1: Hidden.Gem.Score vs Runtime Boxplot

```
plot1 <- ggplot(mydata,aes(x= Runtime, y= Hidden.Gem.Score, fill = Runtime)) +
  stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot() +  ggtitle("Plot 1") + theme_bw()
plot1
```

The above lines of were used to generate a boxplot of hidden gem scores against runtime. As can be seen from the boxplot (figure 1), the median is approximately equal – with > 2 hour movies having slightly higher hidden gem score than others - for movies of all 4 different runtimes. However, the hidden gem score variance for these movies are quite high except 30-60

mins runtime (mainly because there are very few movies of this runtime category). Thus, it can be safely stated that runtime and hidden gem score do not have a strong association.

```
28 testdata <- mydata %>% mutate(FirstLanguage = sub(",.*", "", mydata$Languages))
29
30 plot2 <- ggplot(testdata,aes(x= Hidden.Gem.Score, y= FirstLanguage, fill = FirstLanguage))  +
31         stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot() + ggtitle("Plot 2") + theme_bw()
32 plot2
33
34 langGem_summaries <- with(testdata, tapply(Hidden.Gem.Score, data.frame(FirstLanguage), mean))
35
36 langGem_summaries <- sort(langGem_summaries, decreasing = TRUE)
37 langGem_summaries <- langGem_summaries %>% kable(col.names = "Mean")
38
39 langRepeat <- table(testdata$FirstLanguage)
40 langRepeat <- sort(langRepeat, decreasing = TRUE)
41 langRepeat <- langRepeat %>% kable()
42
43 lang_n_Gem <- testdata %>% select(FirstLanguage, Hidden.Gem.Score)
44
45 mean(lang_n_Gem$Hidden.Gem.Score)
46
47 lang_n_Gem <- lang_n_Gem %>% mutate(Score_Range = cut(Hidden.Gem.Score, c(0, 3.55, 9.2))) %>% group_by(FirstLanguage)
48
49 lang_n_Gem_array <- xtabs(~FirstLanguage+Score_Range, data = lang_n_Gem)
50 lang_n_Gem_array
51
52
53 column_props <- apply(lang_n_Gem_array, c("Score_Range", "FirstLanguage"), sum) %>% prop.table(.,c(2))
54
55 plot3 <- barplot(column_props, col = lang_n_Gem$Score_Range, las = 2, cex.names = 0.6)
56 legend("topright", fill = lang_n_Gem$Score_Range, legend = levels(factor(lang_n_Gem$Score_Range)), title = "Score Range")
57
```

Line 28 in the above code was firstly used to create a column named "FirstLanguage" which only lists the first language in the column "Languages" for each row. For example, if the column entry for "Languages" is "Spanish, German" – the FirstLanguage column entry is "Spanish." This was done under the assumption that the language listed first was the main language the movie was available in – and to simplify the process of finding the association between Languages and Hidden Gem score.

Line 30-32 generates the boxplot of figure 2. To back this boxplot, lines 34-37 generate the leftmost table of figure 3 – which shows the mean hidden gem score based on the 'FirstLanguage' of each movie in a descending order. As can be seen, Tibetan movies had the highest mean at 8.9 while Chinese movies had the lowest mean at 1.7 – a similar result can be seen from the median in the boxplot of figure 2.

However, since the number of movies in each 'FirstLanguage' is not equal, the middle table of figure 3 is included to show what the frequency of each 'FirstLanguage' is in the dataset using lines 39-41 in the code.

Next, the mean Hidden Gem score is found out to be approximately 3.55 using line 45 for the entire dataset – and then the rightmost table of figure 3 is generated to show how many movies of each first language are above average and how many are below average by creating a 'cutoff' point at 3.55 (using lines 47-50 of the code).

As we can see - Arabic, French, Japanese, etc. movies are more likely to receive a higher than average Hidden Gem Score, while English and Thai movies are more likely to receive a lower than average Hidden Gem Score. This data is standardized to a proportion using lines 53-56 of

the code and shown in figure 4. Thus, it would NOT be wrong to claim that movies of particular 'FirstLanguage' like Arabic receive higher Hidden Gem Score than movies of other 'Firstlanguage' like English. This claim is nevertheless partially weakened due to the fact that English has an incredibly high number of entries in the column of 'FirstLanguage' compared to other languages in the column.

```
61 testdata <- testdata %>% mutate(Number_Of_Languages = (str_count(mydata$Languages, ',') + 1))
62
63 testdata <- testdata %>% mutate(Num_Languages = as.character(testdata$Number_Of_Languages))
64
65 class(testdata$Num_Languages)
66
67 view(testdata)
68
69 plot4 <- ggplot(testdata,aes(x= Num_Languages, y= Hidden.Gem.Score, fill = Num_Languages)) +
70   stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot() +  ggtitle("Plot 4") + theme_bw()
71 plot4
72
73 testdata %>% group_by(Num_Languages) %>% summarise(Mean = mean(Hidden.Gem.Score))
```

Furthermore, to evaluate an association between the number of Languages a movie is offered in (for example, if a movie is offered in "English, Spanish" then Num_Languages is 2) and the Hidden Gem Score of the movie, lines 61-63 were written in the above chunk of code.

Finally, lines 69-71 were used to illustrate the relationship between these two attributes – which is shown in figure 5 below. Line 73 was written to find the mean for each value of Num_Languages, which is also included in figure 5.

As can be seen, both the mean and median for each of these are approximately equal, with a comparatively low mean for '10' and comparatively high median for '9'. However, these are not sufficient to conclude that there is a strong association between the number of languages a movie is offered in and the hidden gem score.

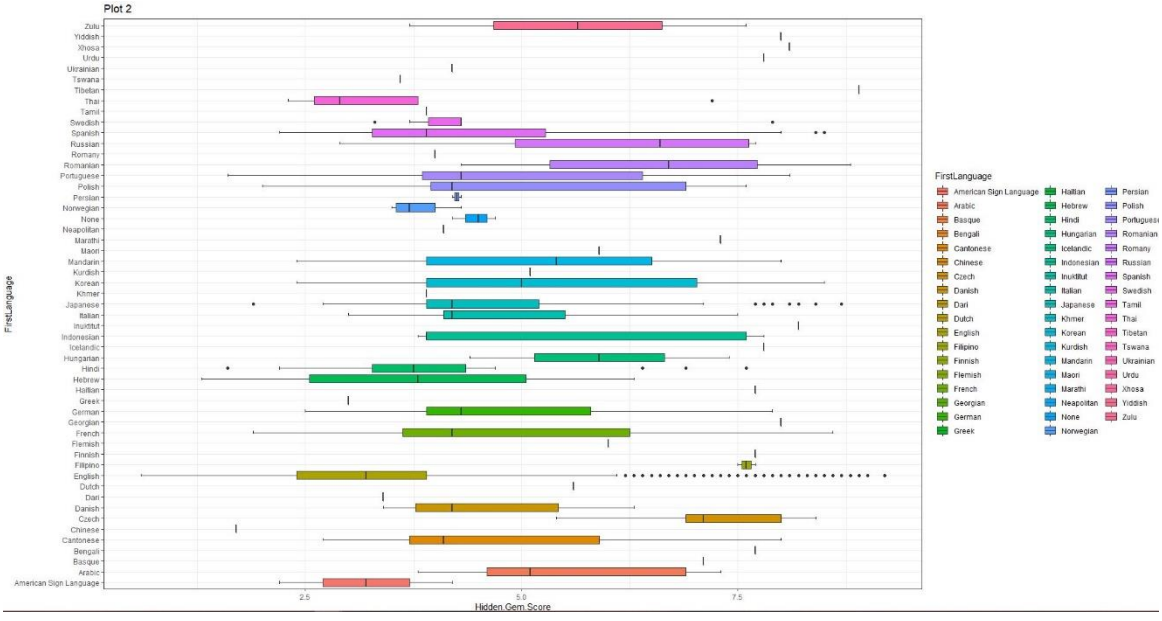


Figure 2: FirstLanguage vs Hidden Gem Score Boxplot

	Mean
Tibetan	8.900000
Inuktitut	8.200000
Xhosa	8.100000
Georgian	8.000000
Yiddish	8.000000
Icelandic	7.800000
Urdu	7.800000
Bengali	7.700000
Finnish	7.700000
Haitian	7.700000
Filipino	7.600000
Marathi	7.300000
Czech	7.242857
Basque	7.100000
Romanian	6.516667
Flemish	6.000000
Russian	5.950000
Hungarian	5.900000
Maori	5.900000
Zulu	5.650000
Arabic	5.600000
Dutch	5.600000
Indonesian	5.400000
Korean	5.318182
Mandarin	5.243902
Kurdish	5.100000
Polish	5.071429
Portuguese	4.973333
French	4.842308
Japanese	4.821538
German	4.781818
Italian	4.747059
Cantonese	4.738095
Danish	4.550000
Spanish	4.540909
Swedish	4.512500
None	4.466667
Persian	4.250000
Ukrainian	4.200000
Neapolitan	4.100000
Hindi	4.016667
Romany	4.000000
Khmer	3.900000
Tamil	3.900000
Hebrew	3.800000
Norwegian	3.800000
Thai	3.760000
Tswana	3.600000
Dari	3.400000
English	3.334644
American Sign Language	3.200000
Greek	3.000000
Chinese	1.700000

Var1	Freq
English	3178
French	78
Japanese	65
Korean	44
Spanish	44
Mandarin	41
Hindi	24
German	22
Cantonese	21
Italian	17
Portuguese	15
Romanian	12
Danish	10
Swedish	8
Arabic	7
Czech	7
Norwegian	7
Polish	7
Indonesian	5
Thai	5
Russian	4
None	3
American Sign Language	2
Filipino	2
Hebrew	2
Hungarian	2
Persian	2
Zulu	2
Basque	1
Bengali	1
Chinese	1
Dari	1
Dutch	1
Finnish	1
Flemish	1
Georgian	1
Greek	1
Haitian	1
Icelandic	1
Inuktitut	1
Khmer	1
Kurdish	1
Maori	1
Marathi	1
Neapolitan	1
Romany	1
Tamil	1
Tibetan	1
Tswana	1
Ukrainian	1
Urdu	1
Xhosa	1
Yiddish	1

	Score_Range	
FirstLanguage	(0,3.55]	(3.55,9.2]
American Sign Language	1	1
Arabic	0	7
Basque	0	1
Bengali	0	1
Cantonese	4	17
Chinese	1	0
Czech	0	7
Danish	2	8
Dari	1	0
Dutch	0	1
English	2023	1155
Filipino	0	2
Finnish	0	1
Flemish	0	1
French	15	63
Georgian	0	1
German	3	19
Greek	1	0
Haitian	0	1
Hebrew	1	1
Hindi	8	16
Hungarian	0	2
Icelandic	0	1
Indonesian	0	5
Inuktitut	0	1
Italian	1	16
Japanese	9	56
Khmer	0	1
Korean	4	40
Kurdish	0	1
Mandarin	6	35
Maori	0	1
Marathi	0	1
Neapolitan	0	1
None	0	3
Norwegian	2	5
Persian	0	2
Polish	1	6
Portuguese	3	12
Romanian	0	12
Romany	0	1
Russian	1	3
Spanish	14	30
Swedish	1	7
Tamil	0	1
Thai	3	2
Tibetan	0	1
Tswana	0	1
Ukrainian	0	1
Urdu	0	1
Xhosa	0	1
Yiddish	0	1
Zulu	0	2

Figure 3 [left to right] : mean by FirstLanguage -> frequency of FirstLanguage -> Score Range of FirstLanguage

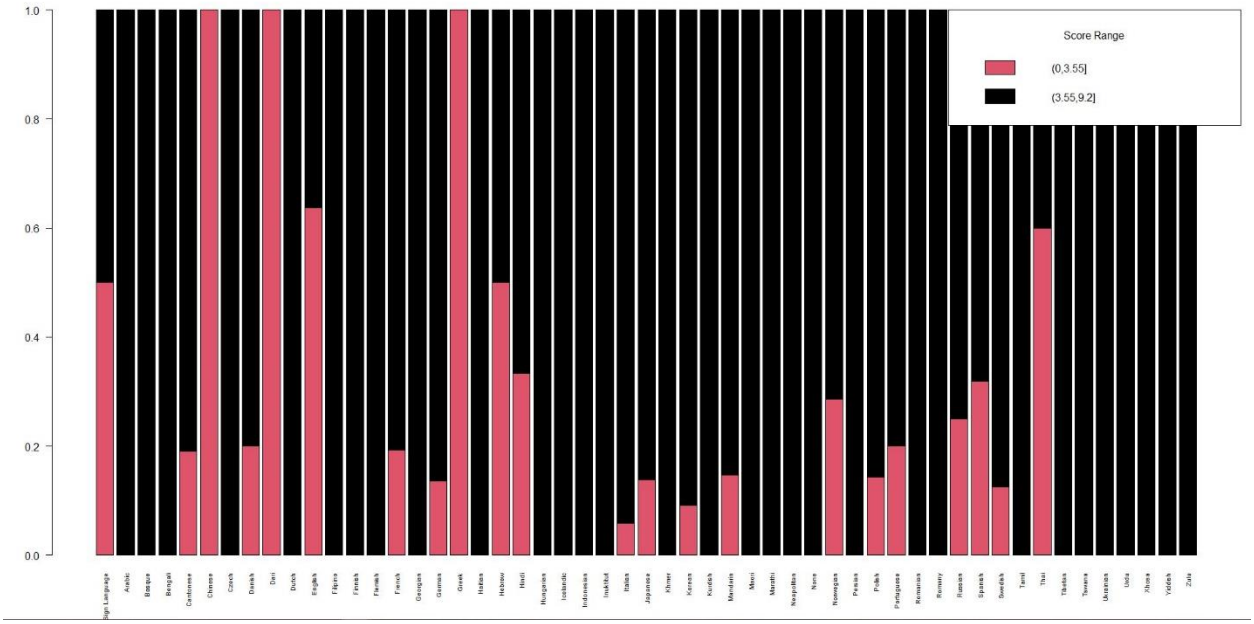
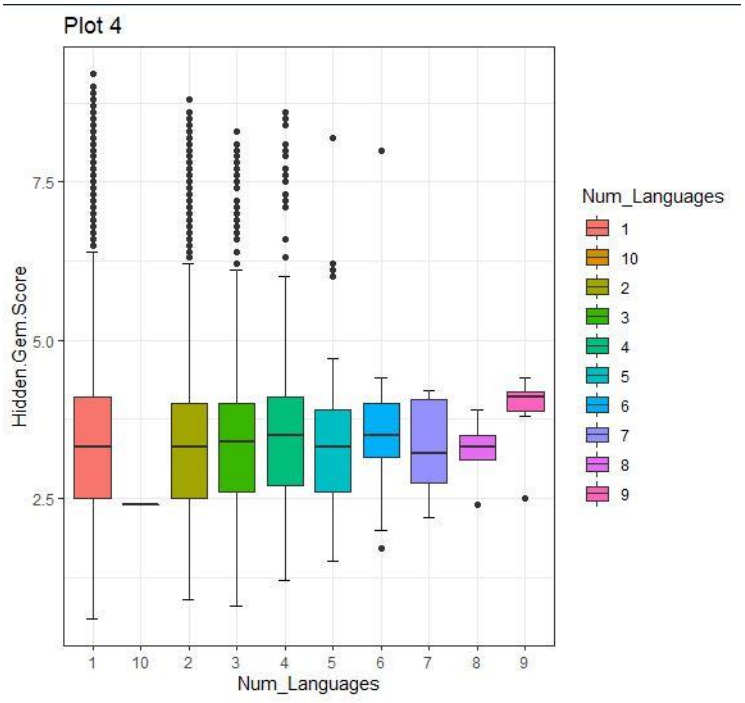


Figure 4: Barplot for numberOfMovies receiving 'lower than average' and 'higher than average' hidden gem score for each 'FirstLanguage'



Num_Languages	Mean
<chr>	<dbl>
1	3.59
10	2.4
2	3.46
3	3.56
4	3.72
5	3.36
6	3.53
7	3.26
8	3.24
9	3.85

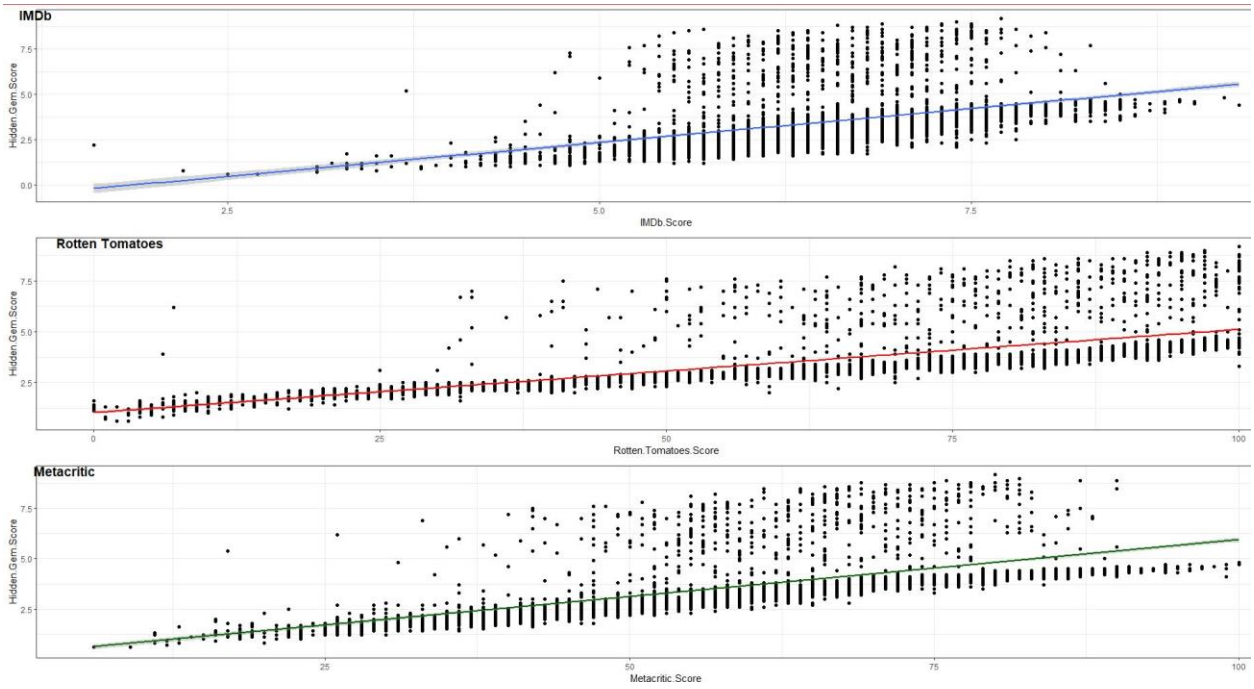
Figure 5 [left to right] : Boxplot of Hidden Gem Score against Num_Languages -> Mean for the number of languages movies are offered in

Part b.

```
80  plot5 <- ggplot(mydata,aes(x= IMDb.Score, y= Hidden.Gem.Score)) + geom_point() + geom_smooth(method = "lm") + theme_bw()
81
82  plot6 <- ggplot(mydata,aes(x= Rotten.Tomatoes.Score, y= Hidden.Gem.Score)) + geom_point() + geom_smooth(method = "lm", col = "red") + theme_bw()
83
84  plot7 <- ggplot(mydata,aes(x= Metacritic.Score, y= Hidden.Gem.Score)) + geom_point() + geom_smooth(method = "lm", col = "darkgreen") + theme_bw()
85
86  figure <- ggarrange(plot5,plot6, plot7,
87                      labels = c("IMDb", "Rotten Tomatoes", "Metacritic"),
88                      ncol = 1, nrow = 3)
89  figure
90
```

The lines of code from 80-84 were written to generate 3 different scatterplots of Hidden Gem Score against each of IMDb, Rotten Tomatoes, and Metacritic respectively. The geom_smooth function was used to show the overall trend and correlation between the scores.

Next, lines 86-89 were written to combine them together in a single diagram as shown below. We can see that the correlation between Hidden Gem Score and each of the 3 scores is quite strong – given we overlook some outliers. This can be seen from the nature of the trendline (blue, red, and green respectively). Thus, it is safe to say that Hidden Gem Score has a direct correlation with each of the 3 scores.



Part c.

```
94  testdata <- testdata %>% mutate(Release_Year = sub("-.*", "", testdata$Release.Date))
95
96  testdata <- testdata %>% mutate(Release_Year = as.numeric(testdata$Release_Year))
97
98  testdata <- testdata %>% mutate(Release_Year_Range = cut(Release_Year, c(1920, 1945, 1970, 1995, 2021)))
99
100 testdata <- testdata %>% mutate(ReleaseYear_Range = case_when(Release_Year_Range == "(1.92e+03,1.94e+03]" ~ "1920-1945",
101                                                               Release_Year_Range == "(1.94e+03,1.97e+03]" ~ "1945-1970",
102                                                               Release_Year_Range == "(1.97e+03,2e+03]" ~ "1970-1995",
103                                                               Release_Year_Range == "(2e+03,2.02e+03]" ~ "1995-2020"))
104
105 testdata <- testdata %>% select(Title:FirstLanguage, Num_Languages, Release_Year, ReleaseYear_Range)
106
107
108 ReleaseSummaries <- testdata %>% group_by(ReleaseYear_Range, Runtime) %>% summarise(MeanScore = mean(Hidden.Gem.Score))
109
110 ReleaseSummaries <- ReleaseSummaries %>% kable()
111
112 ReleaseSummaries
```

Lines 94-105 of the code were written to find the year when each movie was released from its release date, and then categorize the years into 4 segments: 1920-1945, 1945-1970, 1970-1995, 1995-2020

Then, lines 108-112 were used to generate the table shown below which shows the mean Hidden Gem Score for each Release year category based on the runtime of the movies. As we can see, if we consider movies longer than 2 hours to be 'long' movies, the mean Hidden Gem Score has fallen from 4.500 to 3.617. The story is the same for 'moderately long' movies between 1-2 hours whose mean Hidden Gem score has fallen from 4.743 to 3.540.

If the hidden gem score is a representation of how acceptable movies are to users, then the theory that longer movies have gained more acceptance over time is incorrect.

However, if the hidden gem score is a representation of how well the movies are 'hidden', it would mean more users are now watching these longer movies causing the movies to NOT remain 'hidden' anymore and resulting in them having a lower hidden gem score. In that case, the theory proves to be correct.

ReleaseYear_Range	Runtime	MeanScore
1920-1945	> 2 hrs	4.500
1920-1945	1-2 hour	4.743
1945-1970	< 30 minutes	3.900
1945-1970	> 2 hrs	4.177
1945-1970	1-2 hour	4.728
1970-1995	< 30 minutes	3.100
1970-1995	> 2 hrs	3.730
1970-1995	1-2 hour	3.254
1970-1995	30-60 mins	3.500
1995-2020	< 30 minutes	3.764
1995-2020	> 2 hrs	3.617
1995-2020	1-2 hour	3.540