Math 208 H Index Function

2021-12-23

Task:

Find the top 10 directors in the dataset according the Hidden Gem Index (HG-H index) defined as the number of films, H, in the dataset that they have directed which have Hidden Gem Scores that are greater than or equal to H and produce them in a table with their associated HG-H index.

Solution:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(utils)
library(readr)
library(dplyr)

FlixGem <- read_csv("/Users/lilysamuel/Desktop/movie_data.csv")
```

```
## Rows: 15480 Columns: 29
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (21): Title, Genre, Tags, Languages, Series.or.Movie, Country Availabil...
## dbl   (7): Hidden.Gem.Score, IMDb.Score, Rotten.Tomatoes.Score, Metacritic.S...
## date  (1): Netflix.Release.Date
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
FlixGem <- FlixGem %>% drop_na

FlixGem <- FlixGem %>% group_split('Series.or.Movie')

print(FlixGem)
```

```
## <list_of<
##   tbl_df<
##     Title                : character
##     Genre                : character
##     Tags                 : character
##     Languages            : character
##     Series.or.Movie      : character
##     Hidden.Gem.Score     : double
##     Country Availability : character
##     Runtime              : character
##     Director             : character
##     Writer               : character
##     Actors               : character
##     View Rating          : character
##     IMDb.Score           : double
##     Rotten.Tomatoes.Score: double
##     Metacritic.Score     : double
##     Awards Received      : double
##     Awards Nominated For : double
##     Boxoffice            : character
##     Release.Date         : character
##     Netflix.Release.Date : date
##     Production House     : character
##     Netflix Link         : character
##     IMDb Link            : character
##     Summary              : character
##     IMDb Votes           : double
##     Image                : character
##     Poster               : character
##     TMDb Trailer         : character
##     Trailer Site         : character
##     "Series.or.Movie"    : character
##   >
## >[1]>
## [[1]]
## # A tibble: 2,111 x 30
##    Title                 Genre  Tags  Languages Series.or.Movie Hidden.Gem.Score
##    <chr>                 <chr>  <chr> <chr>     <chr>                      <dbl>
##  1 Joker                 Crime~ Dark~ English   Movie                        3.5
##  2 I                     Actio~ Dram~ English,~ Movie                        2.8
##  3 Harrys Daughters      Adven~ Dram~ English   Movie                        4.4
##  4 The Closet            Comedy Kore~ French    Movie                        3.8
##  5 Ordinary People       Drama  Kore~ English   Movie                        4.2
##  6 Stand by Me           Adven~ Kore~ English   Movie                        4.1
##  7 Wonderstruck          Adven~ Chil~ English,~ Movie                        3.6
##  8 The Girl on the Train Crime~ Boll~ English,~ Movie                        2.6
##  9 Red                   Actio~ Dram~ English,~ Movie                        3.4
## 10 Burden                Drama  Movi~ English   Movie                        7.8
## # i 2,101 more rows
## # i 24 more variables: ‘Country Availability‘ <chr>, Runtime <chr>,
## #   Director <chr>, Writer <chr>, Actors <chr>, ‘View Rating‘ <chr>,
## #   IMDb.Score <dbl>, Rotten.Tomatoes.Score <dbl>, Metacritic.Score <dbl>,
## #   ‘Awards Received‘ <dbl>, ‘Awards Nominated For‘ <dbl>, Boxoffice <chr>,
## #   Release.Date <chr>, Netflix.Release.Date <date>, ‘Production House‘ <chr>,
## #   ‘Netflix Link‘ <chr>, ‘IMDb Link‘ <chr>, Summary <chr>, ...
```

```
class(FlixGem)
```

```
## [1] "vctrs_list_of" "vctrs_vctr"    "list"
```

```
FlixGem_Movies <- FlixGem[[1]]

print(FlixGem_Movies)
```

```
## # A tibble: 2,111 x 30
##    Title                 Genre  Tags  Languages Series.or.Movie Hidden.Gem.Score
##    <chr>                 <chr>  <chr> <chr>     <chr>                      <dbl>
##  1 Joker                 Crime~ Dark~ English   Movie                        3.5
##  2 I                     Actio~ Dram~ English,~ Movie                        2.8
##  3 Harrys Daughters      Adven~ Dram~ English   Movie                        4.4
##  4 The Closet            Comedy Kore~ French    Movie                        3.8
##  5 Ordinary People       Drama  Kore~ English   Movie                        4.2
##  6 Stand by Me           Adven~ Kore~ English   Movie                        4.1
##  7 Wonderstruck          Adven~ Chil~ English,~ Movie                        3.6
##  8 The Girl on the Train Crime~ Boll~ English,~ Movie                        2.6
##  9 Red                   Actio~ Dram~ English,~ Movie                        3.4
## 10 Burden                Drama  Movi~ English   Movie                        7.8
## # i 2,101 more rows
## # i 24 more variables: ‘Country Availability‘ <chr>, Runtime <chr>,
## #   Director <chr>, Writer <chr>, Actors <chr>, ‘View Rating‘ <chr>,
## #   IMDb.Score <dbl>, Rotten.Tomatoes.Score <dbl>, Metacritic.Score <dbl>,
## #   ‘Awards Received‘ <dbl>, ‘Awards Nominated For‘ <dbl>, Boxoffice <chr>,
## #   Release.Date <chr>, Netflix.Release.Date <date>, ‘Production House‘ <chr>,
## #   ‘Netflix Link‘ <chr>, ‘IMDb Link‘ <chr>, Summary <chr>, ...
```

```
FlixGem_Movies%>%group_by(Runtime)%>%summarise(hidden_gem_score_avg= mean('Hidden.Gem.Score'))
```

```
## Warning: There were 3 warnings in ‘summarise()‘.
## The first warning was:
## i In argument: ‘hidden_gem_score_avg = mean("Hidden.Gem.Score")‘.
## i In group 1: ‘Runtime = "1-2 hour"‘.
## Caused by warning in ‘mean.default()‘:
## ! argument is not numeric or logical: returning NA
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 2 remaining warnings.
```

```
## # A tibble: 3 x 2
##   Runtime      hidden_gem_score_avg
##   <chr>                       <dbl>
## 1 1-2 hour                       NA
## 2 < 30 minutes                   NA
## 3 > 2 hrs                        NA
```

```
FlixGem_Movie <- separate_rows(FlixGem_Movies, 'Country Availability', sep=", ", convert =TRUE)

class(FlixGem_Movies)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

First, we subset to find a tibble that gives us a list of directors in the dataset, total movies they have directed, and hidden gem score average

```
grouped_movies <-FlixGem_Movies %>% group_by(Director)
print(grouped_movies)
```

```
## # A tibble: 2,111 x 30
## # Groups:   Director [1,120]
##    Title                 Genre  Tags  Languages Series.or.Movie Hidden.Gem.Score
##    <chr>                 <chr>  <chr> <chr>     <chr>                      <dbl>
##  1 Joker                 Crime~ Dark~ English   Movie                        3.5
##  2 I                     Actio~ Dram~ English,~ Movie                        2.8
##  3 Harrys Daughters      Adven~ Dram~ English   Movie                        4.4
##  4 The Closet            Comedy Kore~ French    Movie                        3.8
##  5 Ordinary People       Drama  Kore~ English   Movie                        4.2
##  6 Stand by Me           Adven~ Kore~ English   Movie                        4.1
##  7 Wonderstruck          Adven~ Chil~ English,~ Movie                        3.6
##  8 The Girl on the Train Crime~ Boll~ English,~ Movie                        2.6
##  9 Red                   Actio~ Dram~ English,~ Movie                        3.4
## 10 Burden                Drama  Movi~ English   Movie                        7.8
## # i 2,101 more rows
## # i 24 more variables: ‘Country Availability‘ <chr>, Runtime <chr>,
## #   Director <chr>, Writer <chr>, Actors <chr>, ‘View Rating‘ <chr>,
## #   IMDb.Score <dbl>, Rotten.Tomatoes.Score <dbl>, Metacritic.Score <dbl>,
## #   ‘Awards Received‘ <dbl>, ‘Awards Nominated For‘ <dbl>, Boxoffice <chr>,
## #   Release.Date <chr>, Netflix.Release.Date <date>, ‘Production House‘ <chr>,
## #   ‘Netflix Link‘ <chr>, ‘IMDb Link‘ <chr>, Summary <chr>, ...
```

```
grouped_movies %>% summarise(total_movies_directed= n(),hidden_gem_score_avg = mean('Hidden.Gem.Score')
```

```
## Warning: There were 1120 warnings in ‘summarise()‘.
## The first warning was:
## i In argument: ‘hidden_gem_score_avg = mean("Hidden.Gem.Score")‘.
## i In group 1: ‘Director = "Aaron Katz"‘.
## Caused by warning in ‘mean.default()‘:
## ! argument is not numeric or logical: returning NA
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 1119 remaining warnings.
```

```
## # A tibble: 1,120 x 3
##    Director                        total_movies_directed hidden_gem_score_avg
##    <chr>                                           <int>                <dbl>
##  1 Aaron Katz                                          1                   NA
##  2 Aaron Lieber                                        1                   NA
##  3 Aaron Moorhead, Justin Benson                       1                   NA
##  4 Aaron Sorkin                                        1                   NA
##  5 Aaron Woodley                                       1                   NA
##  6 Abby Kohn, Marc Silverstein                         1                   NA
##  7 Abdellatif Kechiche                                 1                   NA
##  8 Adam Brooks                                         1                   NA
##  9 Adam McKay                                          7                   NA
## 10 Adam Robitel                                        1                   NA
## # i 1,110 more rows
```

Second, we create H index function

```
h_index <- function(input){

    sorted_input <-sort(input, decreasing = F)
    for (i in 1:length(sorted_input)){

      result <- length(sorted_input) - i +1
      if (result <= sorted_input[[i]]){
      return(result)
    }

    }

  return(0)

}

input_test_1<-(c(2,2,4,4,4,4,5)) #just a test

print(h_index(input_test_1))
```

```
## [1] 4
```

We see this function works because there are 5 characters in the vector equal to or greater than 4, therefore the H index should be 4. There test worked.

Now, apply data to the H index function to get H index of directors that we want.

```
x <- separate_rows(FlixGem[[1]],Director,sep=", ",convert =TRUE)
class(x)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
the_director <- unique(x$Director)
class(the_director)
```

```
## [1] "character"
```

```
h_index_vect = c()
number = 0

for (j in the_director){
  number <- number+1
  HG_scores <- x %>% filter(Director==j) %>% select(mean = 'Hidden.Gem.Score')
  h_index_to_use <- h_index(HG_scores[[1]])
  h_index_vect <- c(h_index_vect,h_index_to_use)

}

print(tibble(the_director, director_H_index = h_index_vect) %>% slice_max(h_index_vect, n=10))
```

```
## # A tibble: 18 x 2
##    the_director         director_H_index
##    <chr>                           <dbl>
##  1 Steven Spielberg                    4
##  2 Quentin Tarantino                   4
##  3 Ang Lee                             4
##  4 David Fincher                       4
##  5 Bong Joon Ho                        4
##  6 Woody Allen                         4
##  7 David Mackenzie                     4
##  8 Danny Boyle                         4
##  9 Hayao Miyazaki                      4
## 10 Peter Jackson                       4
## 11 Paul Thomas Anderson                4
## 12 Ridley Scott                        4
## 13 Edgar Wright                        4
## 14 Christopher Nolan                   4
## 15 Steven Soderbergh                   4
## 16 Pedro Almodóvar                     4
## 17 Martin Scorsese                     4
## 18 Alfonso Cuarón                      4
```