Stat 123 Homework 14

Jonathan Wilson

February 13, 2019

```
knitr::opts_knit$set(root.dir =
"C:\\Users\\jon\\Documents\\School\\R\\HW\\HW14")
#Make sure you are calling the right lib in the RIGHT ORDER!

library(plyr); library(magrittr); library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Load the "movies.Rbin" file. Calculate the mean rating for each level of MPAA rating (G, PG, etc.) using this data. Hints: You can download the data in R using the 'load' function and you can access the 'movies' data.frame by loading the R binary file with the 'load' function.

```
load("movies.Rbin")
movies <- data.frame(movies)
#https://stackoverflow.com/questions/27157137/dplyr-only-returning-one-row-
when-using-summarize
movies.mean.rating <- movies %>% group_by(mpaa) %>% summarize(mean.rating =
mean(rating))
movies.mean.rating

## # A tibble: 5 x 2
##   mpaa  mean.rating
##   <chr>       <dbl>
## 1 ""           5.97
## 2 NC-17        5.36
## 3 PG           5.61
## 4 PG-13        5.80
## 5 R            5.42
```

Which year has the most total IMDB votes on the movies made that year?

```
year.most.votes <- movies %>% dplyr::filter(votes == max(votes)) %>%
select(year)
year.most.votes

##   year
## 1 2001

#This does not work like SQL. Each action is executed individually then piped
to the next function as input like in bash. Thus order does not matter.
```

We are interested in finding which year created the most popular movies relative to their budget. Create a function that takes a data.frame and computes the average movie rating and divides it by the average budget.

```
#var <- c(POP=with(df, sum(rating)/ sum(budget)))
pop.movies <- function(df){
  #Turn NA values into 0.
  df$budget[is.na(df$budget)] <- 0
  #Group on year then take mean rating for all movies for that year and
divide by the mean of the budget for a particular year
  pop.df <- df %>% group_by(year) %>% summarize(popularity = mean(rating,
na.rm=TRUE)/mean(budget, na.rm=TRUE))
  #Turn inf values into 0. Values that are essentially zero.
  pop.df$popularity[is.infinite(pop.df$popularity)] <- 0
  return(pop.df)
}
years.pop <- pop.movies(movies)
years.pop <- years.pop[order(years.pop$popularity, decreasing=TRUE), ]
top.year.df <- head(years.pop, 1)
top.year <- top.year.df$year
top.year

## [1] 1906
```

Using your function defined above and the given data, what were the 6 years in which movies were made most efficiently between 1905 and 2005? (Hint: See the Baseball OBP example in Section 11.3.1.)

```
years.pop <- pop.movies(movies)
years.pop <- years.pop[order(years.pop$popularity, decreasing=TRUE), ]
top.year.df <- head(years.pop, 6)
top.six.years <- top.year.df$year
top.six.years

## [1] 1906 1913 1912 1915 1918 1914
```

Thought question: What concerns do you have with this validity of this analysis?

```
#I am not really sure if this a good indicator of how popular a movie is.
Also, much of the data is missing. And another
#Another movie database may have different ratings for movies.
```