HW2_Movie_Ratings notebook

April 12, 2016

1 HW2: The Distribution of Movie Ratings

Movie ratings are numeric scores summarizing the quality of a movie. In this assignment, we study two sources of ratings:

- up-to-date, recently tweeted movie ratings (from MovieTweetings)
- historical movie rating averages (from IMDb).

Ratings from both of these sources are numeric values ranging from 0 to 10.

Perhaps remarkably, the distribution for the two kinds of ratings look the same. The goal of this assignment is to characterize the movie ratings distribution.

> *You are supposed to produce an output file answering four sets of questions (11 questions in all):*
>
> 1. *characterizing the distribution of live MovieTweetings movie ratings*
> 2. *studying the differences between average and median MovieTweetings ratings.*
> 3. *characterizing the distribution of archival IMDb movie ratings*
> 4. *analyzing skewness of the Gamma distribution.*

The details of these questions are laid out in the notebook. And as it explains, your program should produce a CSV file `HW2_output.csv` answering the questions. A correct output file could look like this:

```
lognormal,5.55555,1.11111
skewness,2.22222,
kurtosis,3.33333,
Batman: The Dark Knight,3.33333,8.88888
Batman v Superman: Dawn of Justice,9.11111,9.55555
lognormal,5.22222,1.44444
skewness,2.55555,
kurtosis,3.44444,
1,,
False,,
```

This is just an example of the format of an output file; your output file will be different.

This file characterizes the distribution of ratings as a *lognormal distribution*. This is not correct: the ratings distribution clearly cannot be lognormal, since it is *negatively skewed* (it leans to the right) whereas the lognormal distribution is *positively skewed* (it leans to the left).

Another distribution is needed. The notebook suggests some candidate distributions as possibilities, but you job is to identify one, and obtain the best fit (i.e., maximal likelihood parameters) for the data.

The notebook does not give as much guidance as the earlier assignment notebooks. However, this is also a short assignment. To complete this assignment, please upload two files to CCLE:

1. your output CSV file `HW2_output.csv`

2. your notebook file `HW2_Movie_Ratings.ipynb` (to show your work).

The notebook should have the commands you used to produce the output file. All assignment grading in this course will be automated, so please assume that when uploading files.

We will use Paul Eggert's **Late Policy**: The number of days late is $N = 0$ for the first 24 hrs, $N = 1$ for the next 24 hrs, etc., and if you submit an assignment H hours late, $2^{\lfloor H/24 \rfloor}$ points are deducted.

2 Part 1: Live Movie Ratings – extracted from Tweets

2.1 Live Movie Ratings are available at GitHub

These ratings are updated automatically online by a process scanning current Tweets; see the Movie Tweetings page of Simon Dooms. The information in the tweets has been digested into three tables – about movies, users, and ratings. (Up-to-date snapshots and archives are also available.)

In [142]:
```
URL = "https://raw.githubusercontent.com/sidooms/MovieTweetings/master/latest/ratings.dat"

Ratings = read.table( URL, sep = ":", header=FALSE )[,c(1,3,5,7)]
colnames(Ratings) = c("UserID", "MovieID", "Rating", "TwitterID")

head(Ratings)

# if your connection to github fails when retrieving this dataset, persist in trying
```

Out[142]:

	UserID	MovieID	Rating	TwitterID
1	1	68646	10	1381620027
2	1	113277	10	1379466669
3	2	422720	8	1412178746
4	2	454876	8	1394818630
5	2	790636	7	1389963947
6	2	816711	8	1379963769

In [143]:
```
dim(Ratings)  # not a tiny dataset
```

Out[143]:

1. 489378

2. 4

2.2 Exploration of the Live Ratings

In [144]:
```
# Summary statistics

summary( Ratings$Rating )
```

Out[144]:
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   6.000   7.000   7.304   9.000  10.000
```

In [145]:
```
# Count rating values with table()

CountOfRatings = as.data.frame(table( Ratings$Rating, dnn="rating" ), responseName="count")
CountOfRatings
```

Out[145]:

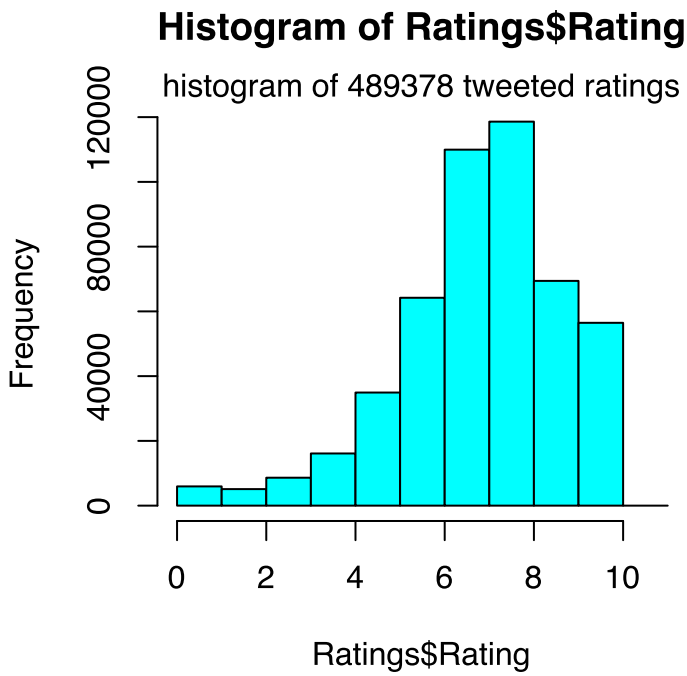
	rating	count
1	0	95
2	1	5844
3	2	5105
4	3	8625
5	4	16114
6	5	34932
7	6	64206
8	7	109957
9	8	118612
10	9	69420
11	10	56468

In [146]:
```
options( repr.plot.width=4, repr.plot.height=4 ) # control plot dimensions
```

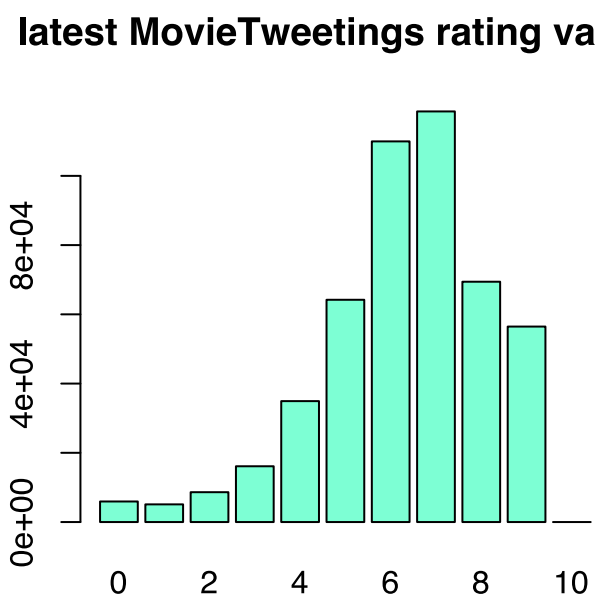
In [147]:
```
# Histogram of Rating values (integer values from 0 to 10)

h = hist( Ratings$Rating, breaks = 0:11, col="cyan" )  # save and plot the histogram

mtext( sprintf("histogram of %d tweeted ratings", length(Ratings$Rating)) )
```



In [148]:
```
barplot( h$counts, names.arg=0:10, col="aquamarine",
                main="latest MovieTweetings rating values" )
```

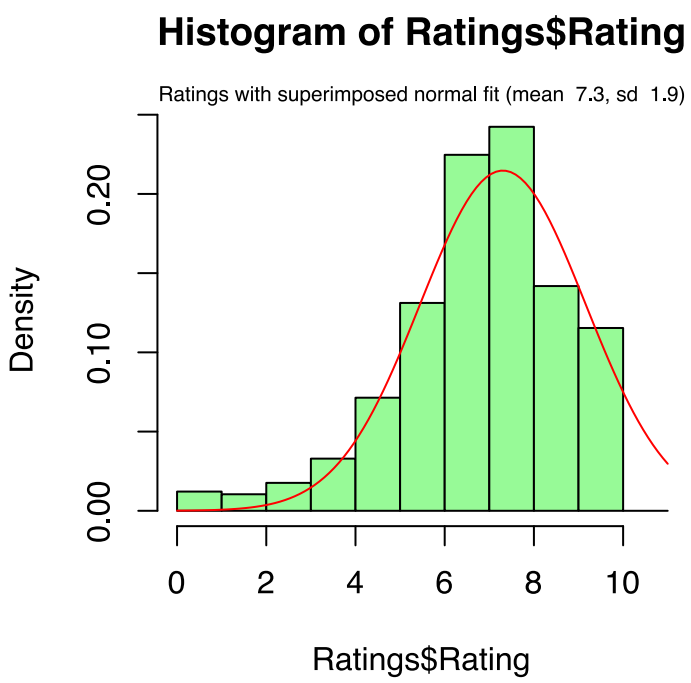


In [149]:
```
hist( Ratings$Rating, probability=TRUE, col="palegreen", breaks=0:11 )

rating_avg = mean(Ratings$Rating)
rating_sd  = sd(Ratings$Rating)

curve( dnorm(x, mean=rating_avg, sd=rating_sd),
      col="red", add=TRUE )  #  add a curve with the normal MLE

mtext(sprintf("Ratings with superimposed normal fit (mean %4.1f, sd %4.1f)",
              rating_avg, rating_sd), cex=0.65)
```



3 Problem 1: Find a distribution that fits the histogram of MovieTweetings Rating values

Question 1: Give the name of a specific distribution (pdf), with maximum likelihood parameter values, that resembles the MovieTweetings Rating values (as closely as you can). To permit distributions like the Beta distribution to be considered, you can scale the rating values. For example, dividing the values by 10 puts them in the interval [0,1], as the Beta distribution requires.

Question 2: determine the skewness of the MovieTweetings ratings.

Question 3: determine the (excess) kurtosis of the MovieTweetings ratings.

The skewness and excess kurtosis values ought to be near zero if the data is normally distributed. Inspecting them is a simple check of whether the data follows a normal distribution.

3.1 You can use the fitdistr() function in this assignment

In [150]:
```
not.installed = function(package_name)  !is.element(package_name, installed.packages()[,1])
if (not.installed("MASS")) install.packages("MASS")

library(MASS)

# example(fitdistr)  #  run examples showing use of the fitdistr() function
```

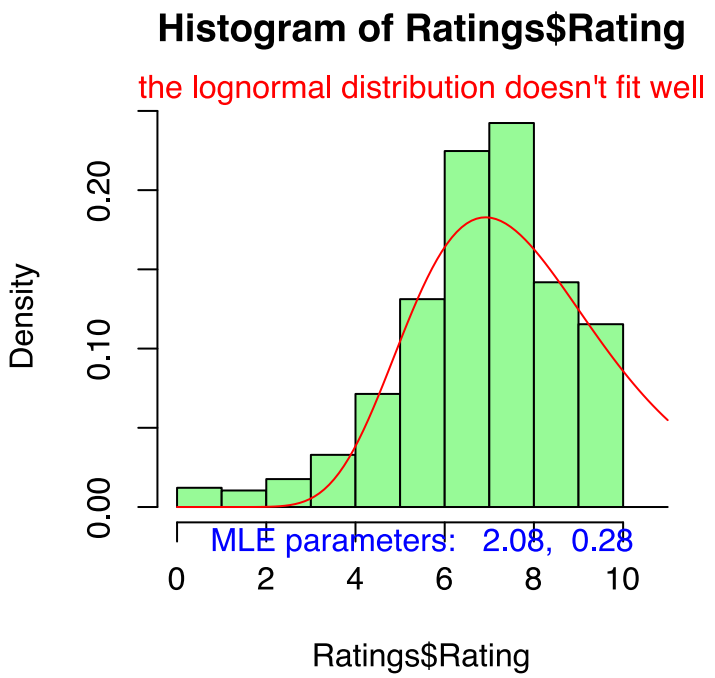
In [151]:
```
# A start at analysis, which needs work:

hist( Ratings$Rating, probability=TRUE, col="palegreen", breaks=0:11 )
```

```
theta = fitdistr( Ratings$Rating+1, "lognormal" )

curve( dlnorm(x+0.5, meanlog=theta$estimate[1], sdlog=theta$estimate[2]),
      col="red", add=TRUE )  #  add a curve with the MLE fit for the lognormal density

mtext( "the lognormal distribution doesn't fit well", col="red" )
mtext( sprintf("MLE parameters:  %5.2f, %5.2f",
               theta$estimate[1], theta$estimate[2]), side=1, col="blue")
```



3.2 "Trending" Movies: movies with more than 50 current ratings

In [152]:
```
NumberOfRatings = data.frame(aggregate( Rating ~ MovieID, length, data=Ratings ))
colnames(NumberOfRatings) = c("MovieID", "NumberOfRatings")

TrendingMovies = subset( NumberOfRatings, NumberOfRatings > 50 )

head(TrendingMovies)
```

Out[152]:

	MovieID	NumberOfRatings
225	21749	51
235	22100	51
451	27977	60
555	31381	87
590	32138	61
655	33467	83

In [153]:
```
nrow(TrendingMovies)
```

Out[153]:
1615

In [154]:
```
max(TrendingMovies$NumberOfRatings)
```

Out[154]:
2951

3.3 Averaged Ratings of Trending Movies

In [155]:
```
RatingsOfTrendingMovies = merge( Ratings, TrendingMovies, by="MovieID" )  # join of tables
head(RatingsOfTrendingMovies)
```

Out[155]:

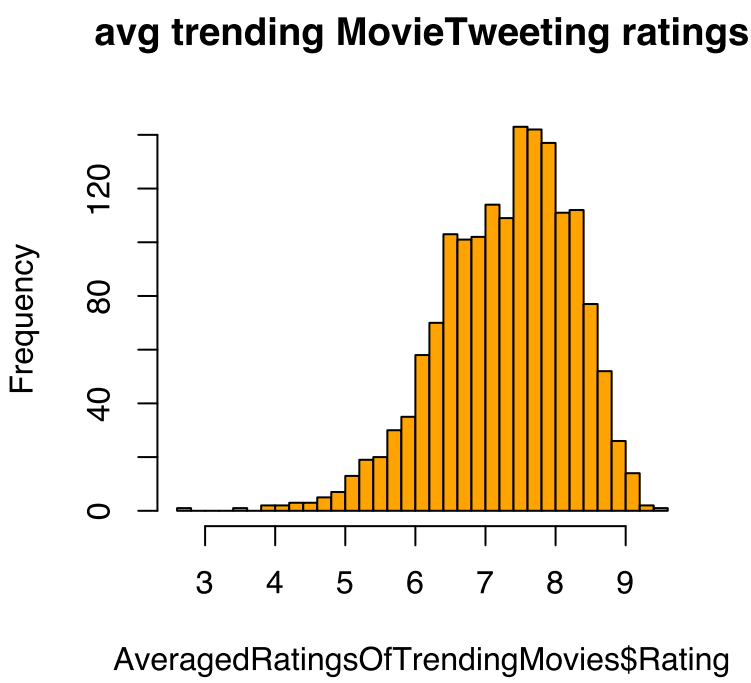
	MovieID	UserID	Rating	TwitterID	NumberOfRatings
1	21749	31658	7	1411560342	51
2	21749	14766	10	1408217153	51
3	21749	3724	9	1419690175	51
4	21749	13477	10	1455471595	51
5	21749	2564	10	1460041235	51
6	21749	4577	8	1422001419	51

In [156]:
```
AveragedRatingsOfTrendingMovies = aggregate( Rating ~ MovieID, mean,
data=RatingsOfTrendingMovies )
head(AveragedRatingsOfTrendingMovies)

hist(AveragedRatingsOfTrendingMovies$Rating, breaks=25,
     col="orange", main="avg trending MovieTweeting ratings")
```

Out[156]:

	MovieID	Rating
1	21749	8.705882
2	22100	8.529412
3	27977	8.633333
4	31381	8.804598
5	32138	8.721311
6	33467	8.662651




```
In [157]: maxAveragedRating = max(AveragedRatingsOfTrendingMovies$Rating)
          maxAveragedRating
```

```
Out[157]:
  9.43649635036496
```

```
In [158]: HotTrendingMovies = subset(AveragedRatingsOfTrendingMovies, Rating == maxAveragedRating)
          HotTrendingMovies
```

Out[158]:

	MovieID	Rating
188	111161	9.436496

4 Download the corresponding MovieTweetings movie Name & Genre information

```
In [159]: URL = "https://raw.githubusercontent.com/sidooms/MovieTweetings/master/latest/movies.dat"
          MovieText = readLines( URL )

          Movies = matrix( sapply( MovieText,
                         function(x) unlist(strsplit(sub(" [(]([0-9]+)[)]", "::\\1",x),"::"))[1:4] ),
                         nrow=length(MovieInformation), ncol=4, byrow=TRUE )
          colnames(Movies) = c("MovieID", "MovieTitle", "Year", "Genres")

          head(Movies)
```

Out[159]:

MovieID	MovieTitle	Year	Genres
0000008	Edison Kinetoscopic Record of a Sneeze	1894	Documentary—Short
0000010	La sortie des usines Lumi¡c3¿¡a8¿re	1895	Documentary—Short
0000012	The Arrival of a Train	1896	Documentary—Short
0000091	Le manoir du diable	1896	Short—Horror
0000417	Le voyage dans la lune	1902	Short—Adventure—Fantasy
0000439	The Great Train Robbery	1903	Short—Action—Crime

4.1 joining the Ratings and Trending Movie information

```
In [160]: Ratings_and_Movies = merge( RatingsOfTrendingMovies, Movies, by="MovieID" )

          head(Ratings_and_Movies)
```

Out[160]:

	MovieID	UserID	Rating	TwitterID	NumberOfRatings	MovieTitle	Year	Genres
1	1001526	32702	6	1453606362	59	Megamind	2010	Animation—Action—Comed
2	1001526	17195	8	1375450650	59	Megamind	2010	Animation—Action—Comed
3	1001526	22716	7	1387735390	59	Megamind	2010	Animation—Action—Comed
4	1001526	2086	7	1367107409	59	Megamind	2010	Animation—Action—Comed
5	1001526	38072	7	1452036480	59	Megamind	2010	Animation—Action—Comed
6	1001526	22420	4	1383345146	59	Megamind	2010	Animation—Action—Comed

5 Problem 2: Compare Average vs. Median Rating values for Trending Movies

After computing Average and Median Rating values for each Trending Movie in the MovieTweetings data:

Question 4: find the name of the movie with the highest Average Rating (and also list its Median Rating and Average Rating). If there is more than one such movie, select any one.

Question 5: find the name of the movie with the largest difference |Median Rating - Average Rating| (and also list its Median Rating and Average Rating). If there is more than one such movie, select any one.

For describing a skewed distribution, the median can be more informative than the mean.

```
In [161]: # Hint:

          # ? aggregate

          # ? by
```

6 Part 2. Historical Movie Ratings – from IMDb

In this part we analyze an historical dataset of movies with ratings from IMDb.

```
In [162]: #  source is at: https://github.com/hadley/ggplot2movies

          if (not.installed("ggplot2movies")) install.packages("ggplot2movies")

          library(ggplot2movies)

          data(movies)

          dim(movies)  # also not a tiny dataset
```

Out[162]:

1. 58788
2. 24

In [163]:
```
summary(movies)
```

```
?movies
```

Out[63]:
```
     title                year          length            budget
 Length:58788       Min.   :1893   Min.   :   1.00   Min.   :        0
 Class :character   1st Qu.:1958   1st Qu.:  74.00   1st Qu.:   250000
 Mode  :character   Median :1983   Median :  90.00   Median :  3000000
                    Mean   :1976   Mean   :  82.34   Mean   : 13412513
                    3rd Qu.:1997   3rd Qu.: 100.00   3rd Qu.: 15000000
                    Max.   :2005   Max.   :5220.00   Max.   :200000000
                                                     NA's   :53573
     rating           votes                r1                r2
 Min.   : 1.000   Min.   :     5.0   Min.   :  0.000   Min.   : 0.000
 1st Qu.: 5.000   1st Qu.:    11.0   1st Qu.:  0.000   1st Qu.: 0.000
 Median : 6.100   Median :    30.0   Median :  4.500   Median : 4.500
 Mean   : 5.933   Mean   :   632.1   Mean   :  7.014   Mean   : 4.022
 3rd Qu.: 7.000   3rd Qu.:   112.0   3rd Qu.:  4.500   3rd Qu.: 4.500
 Max.   :10.000   Max.   :157608.0   Max.   :100.000   Max.   :84.500

       r3               r4                r5               r6
 Min.   : 0.000   Min.   :  0.000   Min.   :  0.000   Min.   : 0.00
 1st Qu.: 0.000   1st Qu.:  0.000   1st Qu.:  4.500   1st Qu.: 4.50
 Median : 4.500   Median :  4.500   Median :  4.500   Median :14.50
 Mean   : 4.721   Mean   :  6.375   Mean   :  9.797   Mean   :13.04
 3rd Qu.: 4.500   3rd Qu.:  4.500   3rd Qu.: 14.500   3rd Qu.:14.50
 Max.   :84.500   Max.   :100.000   Max.   :100.000   Max.   :84.50

       r7               r8               r9               r10
 Min.   :  0.00   Min.   :  0.00   Min.   :  0.000   Min.   :  0.00
 1st Qu.:  4.50   1st Qu.:  4.50   1st Qu.:  4.500   1st Qu.:  4.50
 Median : 14.50   Median : 14.50   Median :  4.500   Median : 14.50
 Mean   : 15.55   Mean   : 13.88   Mean   :  8.954   Mean   : 16.85
 3rd Qu.: 24.50   3rd Qu.: 24.50   3rd Qu.: 14.500   3rd Qu.: 24.50
 Max.   :100.00   Max.   :100.00   Max.   :100.000   Max.   :100.00

     mpaa               Action          Animation          Comedy
 Length:58788       Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 Class :character   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
 Mode  :character   Median :0.00000   Median :0.00000   Median :0.0000
                    Mean   :0.07974   Mean   :0.06277   Mean   :0.2938
                    3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
                    Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

     Drama        Documentary         Romance           Short
 Min.   :0.000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.000   Median :0.00000   Median :0.0000   Median :0.0000
 Mean   :0.371   Mean   :0.05906   Mean   :0.0807   Mean   :0.1609
 3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :1.000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
```

Out[63]:

Movie information and user ratings from IMDB.com.movies

The internet movie database, http://imdb.com/, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon. More about information imdb.com can be found online, http://imdb.com/help/show_leaf?about, including information about the data collection process, http://imdb.com/help/show_leaf?infosource.

`movies`

Format: a data frame with 28819 rows and 24 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget (if known) in US dollars
- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- r1-10. Multiplying by ten gives percentile (to nearest 10%) of users who rated this movie a 1.
- mpaa. MPAA rating.
- action, animation, comedy, drama, documentary, romance, short. Binary variables representing if movie was classified as belonging to that genre.

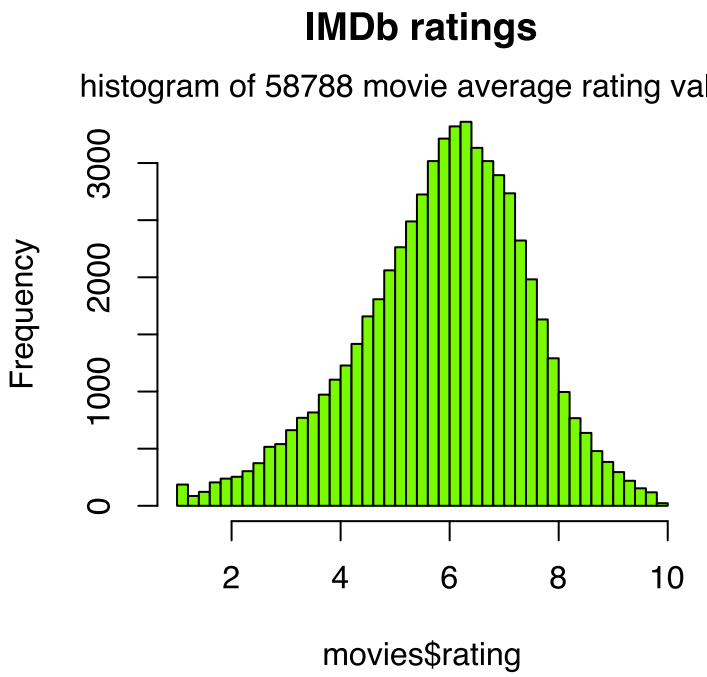
Movies were selected for inclusion if they had a known length and had been rated by at least one imdb user. http://had.co.nz/data/movies/

Out[65]:

	title	year	length	budget	rating	votes	r1	r2	r3	r4	ellip.h	r9	r10	mpaa	
1	$	1971	121	NA	6.4	348	4.5	4.5	4.5	4.5	...	4.5	4.5		
2	$1000 a Touchdown	1939	71	NA	6	20	0	14.5	4.5	24.5	...	4.5	14.5		
3	$21 a Day Once a Month	1941	7	NA	8.2	5	0	0	0	0	...	24.5	24.5		
4	$40,000	1996	70	NA	8.2	6	14.5	0	0	0	...	34.5	45.5		
5	$50,000 Climax Show, The	1975	71	NA	3.4	17	24.5	4.5	0	14.5	...	0	24.5		
6	$pent	2000	91	NA	4.3	45	4.5	4.5	4.5	14.5	...	14.5	14.5		

In [66]:
```
hist( movies$rating, col="lawngreen", main="IMDb ratings", breaks=50 )

mtext( sprintf("histogram of %d movie average rating values", nrow(movies)) )
```



7 Problem 2: Find a distribution that fits the histogram of IMDb Ratings

Question 6: Give the name of a specific distribution (pdf), with maximum likelihood parameter values, that resembles the IMDb Rating values (as closely as you can). Again, to permit distributions like the Beta distribution to be considered, you can scale the rating values. For example, dividing the values by 10 puts them in the interval [0,1], as the Beta distribution requires.

Question 7: determine the skewness of the IMDb ratings.

Question 8: determine the (excess) kurtosis of the IMDb ratings.

7.1 Possibilities: the Extreme Value Distribution, or related distributions

The (Smallest) Extreme Value Distribution models the minimum of a set of values drawn from a single distribution. It is sometimes used for modeling sets of identical independent processes that can fail – and the time of the first failure is the failure time of the entire set. In other words, the distribution models the failure time of the weakest link. The Extreme Value Distribution is related to the Weibull Distribution.

7.2 Possibility: the Beta Distribution

The Beta distribution is a general model for random values of percentages and proportions. It is used very heavily in Bayesian methods. The distribution Beta(α,β) leans to the right when $\alpha > \beta$. Note: As mentioned above, movie rating values are in the interval [0,10] ... The Beta distribution requires all values to be nonnegative, and in the interval [0,1]. Thus the ratings would need to be rescaled here. Warning: the fitdistr() function appears to be fragile when fitting a beta distribution. If you consider the Beta distribution, obtaining MLE parameter values may require a different method.

7.3 Not a Possibility: the Negative Gamma Distribution

The Negative Gamma distribution is a "mirror image" of the Gamma Distribution, defined for $x < 0$ instead of $x > 0$. In other words, the value of the Negative Gamma distribution at $-x$ is defined to be the value of the Gamma distribution at $+x$. Do not consider it in this assignment.

8 Problem 3: Answer two Multiple-Answer questions about the Gamma distribution

Question 9: Give the number of the following expressions that is a formula for the skewness of the Gamma distribution with parameters $\alpha > 0$, $\beta > 0$: (1) α/β (2) $\alpha - \beta$ (3) $\sqrt{\alpha/\beta}$ (4) $\sqrt{\alpha\beta}/2$ (5) $2/\sqrt{\alpha}$ (6) $6/\alpha$ (7) $\log(\alpha)/\beta$ (8) None of the above

Question 10: True or False: the Gamma distribution is not negatively skewed, for any parameter values $\alpha > 0$, $\beta > 0$.

(Again, for this assignment please consider only the usual Gamma Distribution, defined for $x > 0$.) Hint: the skewness measure is the result of an integral for $E[((x - \mu)/\sigma)^3]$, so both of these questions could be answered with a symbolic algebra system, like SymPy or Wolfram Alpha. Some online resources also might give a formula for skewness.

9 Finally: Produce a CSV file "HW2_output.csv" including your answers

9.1 Your output CSV file "HW2_output.csv" should look like this:

```
lognormal,5.55555,1.11111
skewness,2.22222,
kurtosis,3.33333,
Batman: The Dark Knight,3.33333,8.88888
Batman v Superman: Dawn of Justice,9.11111,9.55555
lognormal,5.22222,1.44444
skewness,2.55555,
kurtosis,3.44444,
1,,
False,,
```

If your program had been given the Table above as input, it should print the following CSV file, a table with 10 rows, and three columns:

There should be NO header line in this file. There should be 10 rows, one for each question. Each row should have three fields. You can enter any text description of a distribution, such as lognormal or beta or Beta or whatever. However: the lognormal distribution is just provided as an example; it is not a good description of Rating distributions. Also: do not use the gamma distribution.

10 Submit your output CSV file and your notebook on CCLE.

Upload your .csv file for this assignment, and also upload your .ipynb file (to show your work). Both files are required.