STAT1003

Introduction to Data Science

Solution: Workshop 4

Contents

Wrangling and Exploring Data about Movies

The publicly-available portion of the Internet Movie Database is a popular website that contains virtually all the information you need to know about a movie: year of release, cast, crew, classification, and so on. Another website, The Numbers, contains financial information for a relatively small subset of these films. Data from both these sites is available at the data-aggregation site StatCrunch, and in this workshop, you will be putting two datasets together and carrying out some data-cleaning and exploratory data analysis and visualization.

Data that is available on, or scraped from, the web is encoded in many, many different formats and then has to be imported into R. One of the more convenient ones is a CSV - comma separated value - file. As Baumer *et al.* (2017) write, "[I]t is a non-proprietary comma separated text format that is widely used for data exchange between different software packages. CSV files are easy to understand, but are not compressed, and therefore can take up more space on disk than other formats." The data files you'll be analyzing are CSV files (`IMDB.csv` and `MovieFinances.csv`). Download them from Blackboard into your `I:\STAT1003` folder.

The steps you'll be going through below are simply to 'get to know' the data as a prelude to analyzing it, or to building predictive models. The first step is to get the data into R and then to carry out some simple 'data wrangling' operations; the second is to prepare the data for exploratory analysis (EDA); and the third is to carry out the EDA. Note that there might be some iteration between the second and third steps! There is no one way of doing this: individuals will have their own preferences, and the steps here are only a guide.

In what follows below, we'll be using Windows system commands. In OS X or different flavours of Unix, the system commands are different.

1. Go to the StatCrunch website to find out a little bit more about the content of these files. In the list that you will see, `MovieFinances.csv` comes from the entry entitled "Movie Budgets and Box Office Earnings (Updated Fall 2016)", and `IMDB.csv` from the entry entitled "IMDB Movie Database".

Answer: You should see something that looks like the following:

2. What is the size of these two files?

Answer: Of course, you could check this easily with Windows Explorer, or you could use the command prompt in Windows and then do a directory listing. Nevertheless, you can invoke operating system commands from within *R* as follows:

```
# Windows

# shell('dir *.csv')

# Unix
system("ls -l *.csv")
```

3. Most operating systems will have utilities that will allow you to determine the number of lines in a file (and other information as well) without having to open the file in a program. Google how to do this at the Windows command prompt.

Answer: The command inside the `shell` function is what you'd type at the command prompt, but the chunk below will do the same thing within *R*. (It's a lot easier in Unix or OS X!) As you can see, both files - but especially `IMDB.csv` - contain a lot of records.

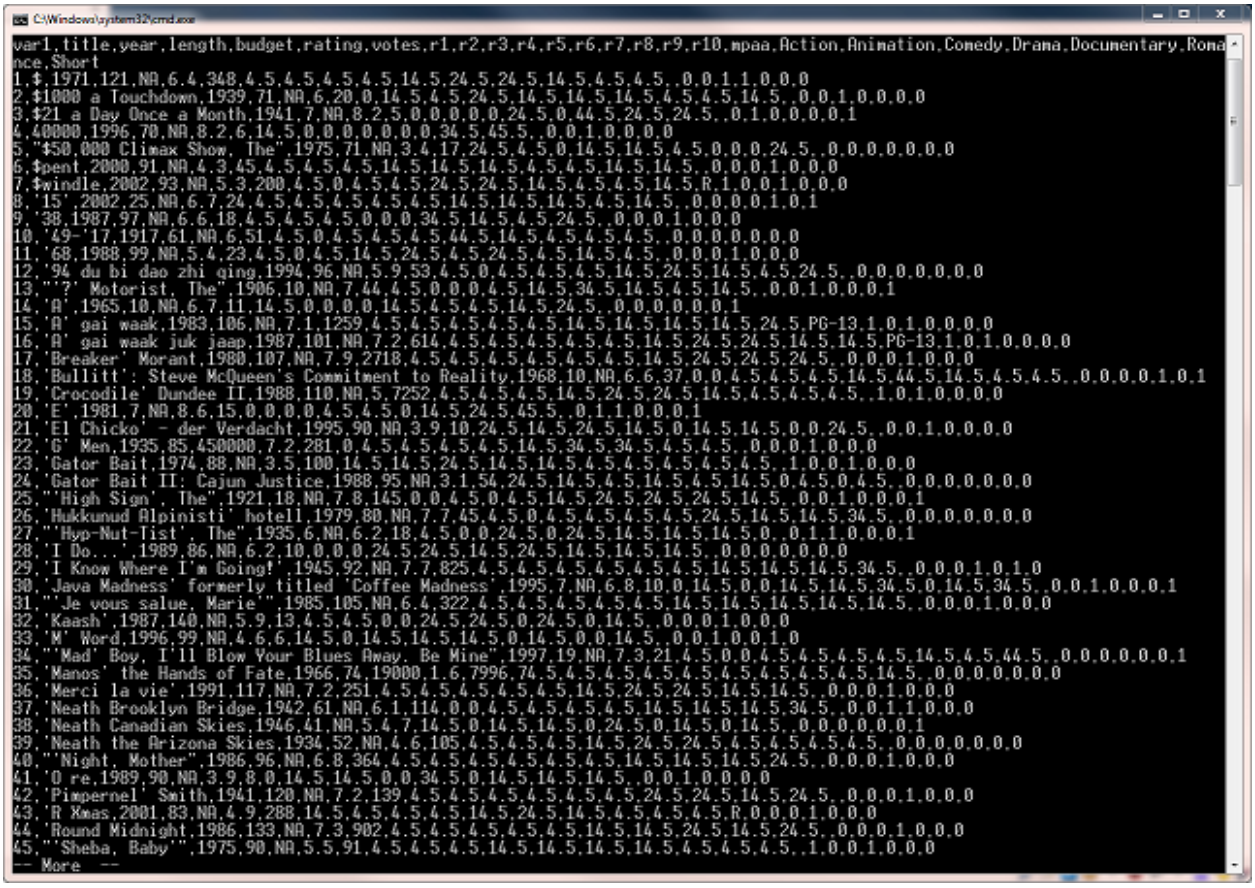
```
# Windows

# shell('find /C /V '' IMDB.csv') shell('find /C /V '' MovieFinances.csv')

# Unix
system("wc -l IMDB.csv")
system("wc -l MovieFinances.csv")
```

4. Before importing a file into R, it's useful to know something about its structure. Again, use Google to find out how to view the contents of a text file at the command line.

Answer: The command to type out the contents of a file at the command line is simply `type filename`, so in this case, `type IMDB.csv` and `type MovieFinances.csv`. If you do so, however, you'll find that the entire file will scroll by very quickly. So, for example, use the command `type IMDB.csv | more` and you'll see something like this:



Note that the first line of the file is a 'header', that is, it contains the variable names.

5. Using the command `read.csv`, read in the data from these two files to create two data frames. For the purposes of this workshop, call them `IMDB` and `MovieFinances`. The values of some of the arguments will depend on the structure of the file that you saw in 3. above.

Answer: Have a look at the help file for the function `read.csv`. It takes lots of arguments, but in this instance, the `.csv` files are well-behaved, so the command is particulary simple; indeed, you don't even need the `header = TRUE` argument because it is the default.

```
MovieFinances <- read.csv("MovieFinances.csv", header = TRUE)
```

```
IMDB <- read.csv("IMDB.csv", header = TRUE)
```

6. After importing the `.csv` files into *R*, have a look at the variable names in both data frames, and then decide whether you need to modify some of them so that they are more compact or more meaningful.

Answer: You could look at the new objects in *RStudio* in the 'Environment' tab, but an alternative is to use the function `colnames`, which extracts and prints the column names.

```
colnames(MovieFinances)
```

```
[1] "Movie"                "Month"                "Day"
[4] "Release.Year"         "Budget..M."           "Domestic.Gross..M."
[7] "Worldwide.Gross..M."
```

```
colnames(IMDB)
```

```
 [1] "var1"        "title"       "year"        "length"      "budget"
 [6] "rating"      "votes"       "r1"          "r2"          "r3"
```

```
[11] "r4"          "r5"          "r6"          "r7"          "r8"
[16] "r9"          "r10"         "mpaa"        "Action"      "Animation"
[21] "Comedy"      "Drama"       "Documentary" "Romance"     "Short"
```

Some of the variable names have changed after the `.csv` file has been imported, e.g., `Budget($M)` has become `Budget..M.`, which is clearly not a very useful name! To change them, we can assign new column names as follows:

```
colnames(MovieFinances)[5] <- "Budget"  # 5th element
colnames(MovieFinances)[6] <- "DomesticGross"
colnames(MovieFinances)[7] <- "WorldwideGross"
```

7. Are there any variables that both datasets have in common? Which ones? Are there any superfluous columns/variables? If there are, remove them.

Answer: The variable `var1` in `IMDB` is superfluous - it is simply the observation number - and so we remove it as follows:

```
IMDB <- IMDB[, -1]  # i.e., remove the first column
```

It appears that the variables `Movie`, `Release.Year`, and `Budget` in `MovieFinances` are the same as `title`, `year`, and `budget` in `IMDB`. Note that we haven't yet checked to see whether their elements are identical for movies that appear in both datasets.

8. Use `head` to look at the first few rows of each data frame, or view them in the *RStudio* data viewer. Do you notice anything that might be unusual?

```
head(MovieFinances)
```

```
                                    Movie Month Day Release.Year Budget
1                                  Avatar   Dec  18         2009    425
2     Star Wars Ep. VII: The Force Awakens   Dec  18         2015    306
3 Pirates of the Caribbean: At World's End   May  24         2007    300
4                                 Spectre   Nov   6         2015    300
5                    The Dark Knight Rises   Jul  20         2012    275
6                         The Lone Ranger   Jul   2         2013    275
  DomesticGross WorldwideGross
1     760.50762      2783.9190
2     936.66223      2058.6622
3     309.42043       963.4204
4     200.07417       879.6209
5     448.13910      1084.4391
6      89.30212       260.0021
```

```
head(IMDB)
```

```
                    title year length budget rating votes   r1   r2  r3   r4
1                       $ 1971    121     NA    6.4   348  4.5  4.5 4.5  4.5
2        $1000 a Touchdown 1939     71     NA    6.0    20  0.0 14.5 4.5 24.5
3   $21 a Day Once a Month 1941      7     NA    8.2     5  0.0  0.0 0.0  0.0
4                   40000 1996     70     NA    8.2     6 14.5  0.0 0.0  0.0
5 $50,000 Climax Show, The 1975     71     NA    3.4    17 24.5  4.5 0.0 14.5
6                   $pent 2000     91     NA    4.3    45  4.5  4.5 4.5 14.5
    r5   r6   r7   r8   r9  r10 mpaa Action Animation Comedy Drama Documentary
1 14.5 24.5 24.5 14.5  4.5  4.5           0         0      1     1           0
2 14.5 14.5 14.5  4.5  4.5 14.5           0         0      1     0           0
3  0.0 24.5  0.0 44.5 24.5 24.5           0         1      0     0           0
4  0.0  0.0  0.0  0.0 34.5 45.5           0         0      1     0           0
5 14.5  4.5  0.0  0.0  0.0 24.5           0         0      0     0           0
```

```
6 14.5 14.5  4.5  4.5 14.5 14.5           0         0      0     1           0
  Romance Short
1       0     0
2       0     0
3       0     1
4       0     0
5       0     0
6       0     0
```

Answer: The variable **Budget:** in `IMDB` has a special symbol, `NA`, which denotes a missing value ('not available'). There may be other variables that have missing values too - we've only looked at the first six rows of two very large datasets!

9. Examine the structure of the variables in each data frame. What do you notice about the type of the variables `title` and `mpaa` in IMDB and `Movie` and `Month` in `MovieFinances`? What should we do about them?

Answer: Recall that the function `str` provides a summary of the data object and the variables it contains.

```
str(IMDB)
```

```
'data.frame':   58786 obs. of  24 variables:
 $ title      : Factor w/ 56005 levels "-30-","...4 ...3 ...2 ...1 ...morte",..: 84 85 86 431 87 88 89 3
 $ year       : int  1971 1939 1941 1996 1975 2000 2002 2002 1987 1917 ...
 $ length     : int  121 71 7 70 71 91 93 25 97 61 ...
 $ budget     : int  NA NA NA NA NA NA NA NA NA NA ...
 $ rating     : num  6.4 6 8.2 8.2 3.4 4.3 5.3 6.7 6.6 6 ...
 $ votes      : int  348 20 5 6 17 45 200 24 18 51 ...
 $ r1         : num  4.5 0 0 14.5 24.5 4.5 4.5 4.5 4.5 4.5 ...
 $ r2         : num  4.5 14.5 0 0 4.5 4.5 0 4.5 4.5 0 ...
 $ r3         : num  4.5 4.5 0 0 0 4.5 4.5 4.5 4.5 4.5 ...
 $ r4         : num  4.5 24.5 0 0 14.5 14.5 4.5 4.5 0 4.5 ...
 $ r5         : num  14.5 14.5 0 0 14.5 14.5 24.5 4.5 0 4.5 ...
 $ r6         : num  24.5 14.5 24.5 0 4.5 14.5 24.5 14.5 0 44.5 ...
 $ r7         : num  24.5 14.5 0 0 0 4.5 14.5 14.5 34.5 14.5 ...
 $ r8         : num  14.5 4.5 44.5 0 0 4.5 4.5 14.5 14.5 4.5 ...
 $ r9         : num  4.5 4.5 24.5 34.5 0 14.5 4.5 4.5 4.5 4.5 ...
 $ r10        : num  4.5 14.5 24.5 45.5 24.5 14.5 14.5 14.5 24.5 4.5 ...
 $ mpaa       : Factor w/ 5 levels "","NC-17","PG",..: 1 1 1 1 1 1 5 1 1 1 ...
 $ Action     : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Animation  : int  0 0 1 0 0 0 0 0 0 0 ...
 $ Comedy     : int  1 1 0 1 0 0 0 0 0 0 ...
 $ Drama      : int  1 0 0 0 0 1 1 0 1 0 ...
 $ Documentary: int  0 0 0 0 0 0 0 1 0 0 ...
 $ Romance    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Short      : int  0 0 1 0 0 0 0 1 0 0 ...
```

```
str(MovieFinances)
```

```
'data.frame':   5222 obs. of  7 variables:
 $ Movie         : Factor w/ 5154 levels "[Rec]","[Rec] 2",..: 387 3549 2913 3494 3924 4249 2053 3684 35
 $ Month         : Factor w/ 12 levels "Apr","Aug","Dec",..: 3 3 9 10 6 6 8 10 9 9 ...
 $ Day           : int  18 18 24 6 20 2 9 24 4 1 ...
 $ Release.Year  : int  2009 2015 2007 2015 2012 2013 2012 2010 2007 2015 ...
 $ Budget        : num  425 306 300 300 275 275 275 260 258 250 ...
 $ DomesticGross : num  761 937 309 200 448 ...
 $ WorldwideGross: num  2784 2059 963 880 1084 ...
```

We can clearly see that the variables `title`, `mpaa`, `Movie`, and `Month` are factor variables, but perhaps `title` and `Movie` don't really need to be. That is a consequence of the way in which the data were imported using the function `read.csv`. We could convert `title` and `Movie` to character variables by using the function `as.character`:

```
IMDB$title <- as.character(IMDB$title)
class(IMDB$title)
```

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

```
MovieFinances$Movie <- as.character(MovieFinances$Movie)
class(MovieFinances$Movie)
```

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

Note that the variable `mpaa` is a factor variable which has multiple levels for the MPAA (Motion Picture Association of America) film classifications. However, one of the levels appears to be empty (`""`), so we need to do something about that. We could replace the empty values by `NA` (missing value), but it's probably better to create a new class name, `"Unknown"`. Here's how we can do that:

```
levels(IMDB$mpaa)  # the first element is what we want to replace
```

```
[1] ""      "NC-17" "PG"    "PG-13" "R"
```

```
levels(IMDB$mpaa)[1] <- "Unknown"
levels(IMDB$mpaa)
```

```
[1] "Unknown" "NC-17"   "PG"      "PG-13"   "R"
```

10. So it looks like we're going to do need to do some manipulation, but perhaps it's better to merge the data sets together because they'll only have some of the same common elements. Have a look at the help file for the function `merge`, and then merge the two datasets together. Call the result `AllData`. What variables should we merge on?

Answer: Make sure you understand the arguments to `merge`, especially if we're going to merge on more than one variable. Because the movie title and release year appear to be in common in these two datasets, let's merge on those. Note that we haven't actually checked whether the titles of the same movies in each dataset have been spelled exactly the same, nor whether the release years of those movies are the same. But that would be something we'd want to do in practice.

```
AllData <- merge(IMDB, MovieFinances, by.x = c("title", "year"), by.y = c("Movie",
    "Release.Year"))
```

11. How large is the merged dataset?

Answer: Using `dim` to determine the size of the dataset, we get

```
dim(AllData)
```

```
[1] 1507   29
```

So, it appears (though we'd need to check that) that only 1507 movie titles (with the same release year) appear in both datasets. The variables in `AllData` are

```
colnames(AllData)
```

```
 [1] "title"          "year"           "length"         "budget"
 [5] "rating"         "votes"          "r1"             "r2"
 [9] "r3"             "r4"             "r5"             "r6"
[13] "r7"             "r8"             "r9"             "r10"
[17] "mpaa"           "Action"         "Animation"      "Comedy"
[21] "Drama"          "Documentary"    "Romance"        "Short"
```

```
[25] "Month"          "Day"            "Budget"         "DomesticGross"
[29] "WorldwideGross"
```

As you might have noticed, the titles in the two datasets are not necessarily in the same format: for example, in `IMDB`, the 1973 movie *The Exorcist* is listed as `Exorcist, The`, but in `MovieFinances` it is listed as `The Exorcist`. So, bear in mind that `AllData` will not contain any films that begin with `The`!

12. Try out the function `summary` using the data frame as the argument. What kind of information does it produce? What unusual aspects do you notice?

The function `summary` gives you the 'five-number summary' along with the mean of each quantitative variable. For factor or character variable, it tabulates the number of each factor level or character string. Note that this is not always useful: for example, for the film-classification variables (`Action`, `Animation`, etc.), which are 0/1 variables, the five-number summary is not particularly useful. Furthermore, summary removes `NA`s from the data before calculating the five-number summary, so we no longer have any information about, for example, how many `NA`s there might be for each variable.

```
summary(AllData)
```

```
   title                year          length          budget
Length:1507        Min.   :1916   Min.   : 66.0   Min.   :     5000
Class :character   1st Qu.:1990   1st Qu.: 95.0   1st Qu.:  8300000
Mode  :character   Median :1998   Median :105.0   Median : 20000000
                   Mean   :1994   Mean   :110.4   Mean   : 31506447
                   3rd Qu.:2002   3rd Qu.:120.0   3rd Qu.: 45000000
                   Max.   :2005   Max.   :320.0   Max.   :200000000
                                                  NA's   :174
    rating          votes             r1              r2
Min.   :1.700   Min.   :     5   Min.   : 0.000   Min.   : 0.000
1st Qu.:5.400   1st Qu.:  2263   1st Qu.: 4.500   1st Qu.: 4.500
Median :6.300   Median :  5839   Median : 4.500   Median : 4.500
Mean   :6.174   Mean   : 10518   Mean   : 6.749   Mean   : 4.927
3rd Qu.:7.100   3rd Qu.: 13158   3rd Qu.: 4.500   3rd Qu.: 4.500
Max.   :8.800   Max.   :132745   Max.   :74.500   Max.   :24.500

      r3              r4              r5              r6
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
1st Qu.: 4.500   1st Qu.: 4.500   1st Qu.: 4.500   1st Qu.:14.50
Median : 4.500   Median : 4.500   Median : 4.500   Median :14.50
Mean   : 5.292   Mean   : 6.463   Mean   : 9.487   Mean   :14.12
3rd Qu.: 4.500   3rd Qu.: 4.500   3rd Qu.:14.500   3rd Qu.:14.50
Max.   :14.500   Max.   :24.500   Max.   :24.500   Max.   :24.50

      r7              r8              r9              r10             mpaa
Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 4.50   Unknown:547
1st Qu.:14.50   1st Qu.: 4.50   1st Qu.: 4.500   1st Qu.: 4.50   NC-17  :  2
Median :14.50   Median :14.50   Median : 4.500   Median : 4.50   PG     :127
Mean   :17.95   Mean   :15.47   Mean   : 9.175   Mean   :11.79   PG-13  :347
3rd Qu.:24.50   3rd Qu.:24.50   3rd Qu.:14.500   3rd Qu.:14.50   R      :484
Max.   :34.50   Max.   :34.50   Max.   :24.500   Max.   :64.50

    Action         Animation          Comedy          Drama
Min.   :0.0000   Min.   :0.00000   Min.   :0.000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.0000
Median :0.0000   Median :0.00000   Median :0.000   Median :0.0000
Mean   :0.2455   Mean   :0.03451   Mean   :0.428   Mean   :0.4831
```

```
 3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.00000   Max.   :1.000   Max.   :1.0000

  Documentary          Romance           Short       Month          Day
 Min.   :0.000000   Min.   :0.0000   Min.   :0   Dec    :202   Min.   : 1.00
 1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0   Oct    :156   1st Qu.: 8.00
 Median :0.000000   Median :0.0000   Median :0   Jun    :137   Median :16.00
 Mean   :0.008626   Mean   :0.1911   Mean   :0   Nov    :132   Mean   :15.56
 3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0   Jul    :128   3rd Qu.:23.00
 Max.   :1.000000   Max.   :1.0000   Max.   :0   Aug    :125   Max.   :31.00
                                                (Other):627
     Budget        DomesticGross    WorldwideGross
 Min.   :  0.007   Min.   :  0.00   Min.   :   0.00
 1st Qu.:  7.100   1st Qu.:  9.00   1st Qu.:  10.59
 Median : 20.000   Median : 26.57   Median :  34.00
 Mean   : 29.754   Mean   : 46.05   Mean   :  83.59
 3rd Qu.: 42.000   3rd Qu.: 60.29   3rd Qu.: 102.17
 Max.   :200.000   Max.   :658.67   Max.   :2207.62
```

13. Unfortunately, `summary` doesn't really tell us about missing values, but the function `describe` in the package `Hmisc` does. Install the package from the `Tools` menu, and then load the library `Hmisc` to be able to use `describe`. Scan the output of `describe` and try to understand what it's telling you.

```
require(Hmisc)
describe(AllData)
```

```
AllData

 29  Variables      1507  Observations
--------------------------------------------------------------------------------
title
       n  missing distinct
    1507        0     1501

lowest : 102 Dalmatians        12 Angry Men          13 Going On 30        15 Minutes            1776
highest: Young Frankenstein    Young Guns            Young Sherlock Holmes Zero Effect           Zoolan
--------------------------------------------------------------------------------
year
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0       75    0.997     1994    12.12     1967     1979
     .25      .50      .75      .90      .95
    1990     1998     2002     2004     2004

lowest : 1916 1925 1927 1930 1931, highest: 2001 2002 2003 2004 2005
--------------------------------------------------------------------------------
length
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0      129        1    110.4    23.64     85.0     88.6
     .25      .50      .75      .90      .95
    95.0    105.0    120.0    135.4    152.0

lowest :  66  67  68  70  73, highest: 233 236 242 259 320
--------------------------------------------------------------------------------
budget
```

```
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1333      174      225        1 31506447 32178763   934000  2000000
     .25      .50      .75      .90      .95
 8300000 20000000 45000000 75000000 90000000

lowest :      5000      7000     10000     22000     23000
highest: 160000000 170000000 175000000 185000000 200000000
---------------------------------------------------------------------------
rating
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0       69    0.999    6.174    1.405      3.9      4.5
     .25      .50      .75      .90      .95
     5.4      6.3      7.1      7.7      8.0

lowest : 1.7 1.9 2.0 2.1 2.3, highest: 8.4 8.5 8.6 8.7 8.8
---------------------------------------------------------------------------
votes
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0     1448        1    10518    12175    383.4    701.8
     .25      .50      .75      .90      .95
  2263.0   5839.0  13158.5  24301.4  36553.1

lowest :      5     15     20     25     31, highest:  97667 100267 109991 112092 132745
---------------------------------------------------------------------------
r1
       n  missing distinct     Info     Mean      Gmd
    1507        0        9    0.359    6.749    4.111

lowest :  0.0  4.5 14.5 24.5 34.5, highest: 34.5 44.5 45.5 64.5 74.5

Value        0.0   4.5  14.5  24.5  34.5  44.5  45.5  64.5  74.5
Frequency      3  1299   134    41    13     7     3     4     3
Proportion 0.002 0.862 0.089 0.027 0.009 0.005 0.002 0.003 0.002
---------------------------------------------------------------------------
r2
       n  missing distinct     Info     Mean      Gmd
    1507        0        4    0.135    4.927   0.8882

Value        0.0   4.5  14.5  24.5
Frequency      6  1436    63     2
Proportion 0.004 0.953 0.042 0.001
---------------------------------------------------------------------------
r3
       n  missing distinct     Info     Mean      Gmd
    1507        0        3    0.233    5.292    1.525

Value        0.0   4.5  14.5
Frequency      6  1379   122
Proportion 0.004 0.915 0.081
---------------------------------------------------------------------------
r4
       n  missing distinct     Info     Mean      Gmd
    1507        0        4     0.48    6.463    3.215
```

```
Value        0.0   4.5  14.5  24.5
Frequency      5  1206   294     2
Proportion 0.003 0.800 0.195 0.001
-------------------------------------------------------------------------------
r5
       n  missing distinct     Info     Mean      Gmd
    1507        0        4    0.765    9.487    5.523

Value        0.0   4.5  14.5  24.5
Frequency      1   795   670    41
Proportion 0.001 0.528 0.445 0.027
-------------------------------------------------------------------------------
r6
       n  missing distinct     Info     Mean      Gmd
    1507        0        4    0.815    14.12    7.036

Value        0.0   4.5  14.5  24.5
Frequency      1   371   820   315
Proportion 0.001 0.246 0.544 0.209
-------------------------------------------------------------------------------
r7
       n  missing distinct     Info     Mean      Gmd
    1507        0        5    0.837    17.95     7.41

lowest :  0.0  4.5 14.5 24.5 34.5, highest:  0.0  4.5 14.5 24.5 34.5

Value        0.0   4.5  14.5  24.5  34.5
Frequency      3   170   672   629    33
Proportion 0.002 0.113 0.446 0.417 0.022
-------------------------------------------------------------------------------
r8
       n  missing distinct     Info     Mean      Gmd
    1507        0        5    0.886    15.47    8.739

lowest :  0.0  4.5 14.5 24.5 34.5, highest:  0.0  4.5 14.5 24.5 34.5

Value        0.0   4.5  14.5  24.5  34.5
Frequency      1   396   593   491    26
Proportion 0.001 0.263 0.393 0.326 0.017
-------------------------------------------------------------------------------
r9
       n  missing distinct     Info     Mean      Gmd
    1507        0        4     0.74    9.175    6.258

Value        0.0   4.5  14.5  24.5
Frequency      1   925   457   124
Proportion 0.001 0.614 0.303 0.082
-------------------------------------------------------------------------------
r10
       n  missing distinct     Info     Mean      Gmd
    1507        0        7    0.833    11.79    9.145

lowest :  4.5 14.5 24.5 34.5 44.5, highest: 24.5 34.5 44.5 45.5 64.5
```

```
Value        4.5  14.5  24.5  34.5  44.5  45.5  64.5
Frequency    769   481   177    59    16     4     1
Proportion 0.510 0.319 0.117 0.039 0.011 0.003 0.001
---------------------------------------------------------------------------
mpaa
       n  missing distinct
    1507        0        5

lowest : Unknown NC-17   PG      PG-13   R
highest: Unknown NC-17   PG      PG-13   R

Value      Unknown   NC-17      PG   PG-13       R
Frequency      547       2     127     347     484
Proportion   0.363   0.001   0.084   0.230   0.321
---------------------------------------------------------------------------
Action
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2    0.556      370   0.2455   0.3707

---------------------------------------------------------------------------
Animation
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2      0.1       52  0.03451  0.06667

---------------------------------------------------------------------------
Comedy
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2    0.734      645    0.428     0.49

---------------------------------------------------------------------------
Drama
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2    0.749      728   0.4831   0.4998

---------------------------------------------------------------------------
Documentary
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2    0.026       13 0.008626  0.01712

---------------------------------------------------------------------------
Romance
       n  missing distinct     Info      Sum     Mean      Gmd
    1507        0        2    0.464      288   0.1911   0.3094

---------------------------------------------------------------------------
Short
       n  missing distinct     Info     Mean      Gmd
    1507        0        1        0        0        0

Value         0
Frequency  1507
Proportion    1
---------------------------------------------------------------------------
Month
```

```
       n  missing distinct
    1507        0       12

lowest : Apr Aug Dec Feb Jan, highest: Mar May Nov Oct Sep

Value        Apr   Aug   Dec   Feb   Jan   Jul   Jun   Mar   May   Nov   Oct
Frequency     99   125   202   103   103   128   137   106    96   132   156
Proportion 0.066 0.083 0.134 0.068 0.068 0.085 0.091 0.070 0.064 0.088 0.104

Value        Sep
Frequency    120
Proportion 0.080
---------------------------------------------------------------------------
Day
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0       31    0.999    15.56    10.12        1        3
     .25      .50      .75      .90      .95
       8       16       23       27       29

lowest :  1  2  3  4  5, highest: 27 28 29 30 31
---------------------------------------------------------------------------
Budget
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0      246        1    29.75    30.84      1.0      2.0
     .25      .50      .75      .90      .95
     7.1     20.0     42.0     73.0     90.0

lowest :   0.007   0.023   0.025   0.042   0.050
highest: 151.500 155.000 170.000 175.000 200.000
---------------------------------------------------------------------------
DomesticGross
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0     1449        1    46.05    53.06   0.5572   2.1409
     .25      .50      .75      .90      .95
  9.0000  26.5700  60.2903 115.9288 162.4071

lowest :   0.000000   0.004655   0.006260   0.007680   0.009598
highest: 380.529370 395.708305 403.706375 441.226247 658.672302
---------------------------------------------------------------------------
WorldwideGross
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1507        0     1456        1    83.59    108.4    0.717    2.568
     .25      .50      .75      .90      .95
  10.595   34.004  102.171  232.970  335.800

lowest :    0.000000    0.004655    0.006260    0.007680    0.009598
highest:  878.979634  936.429370  937.008132 1038.812584 2207.615668
---------------------------------------------------------------------------
```

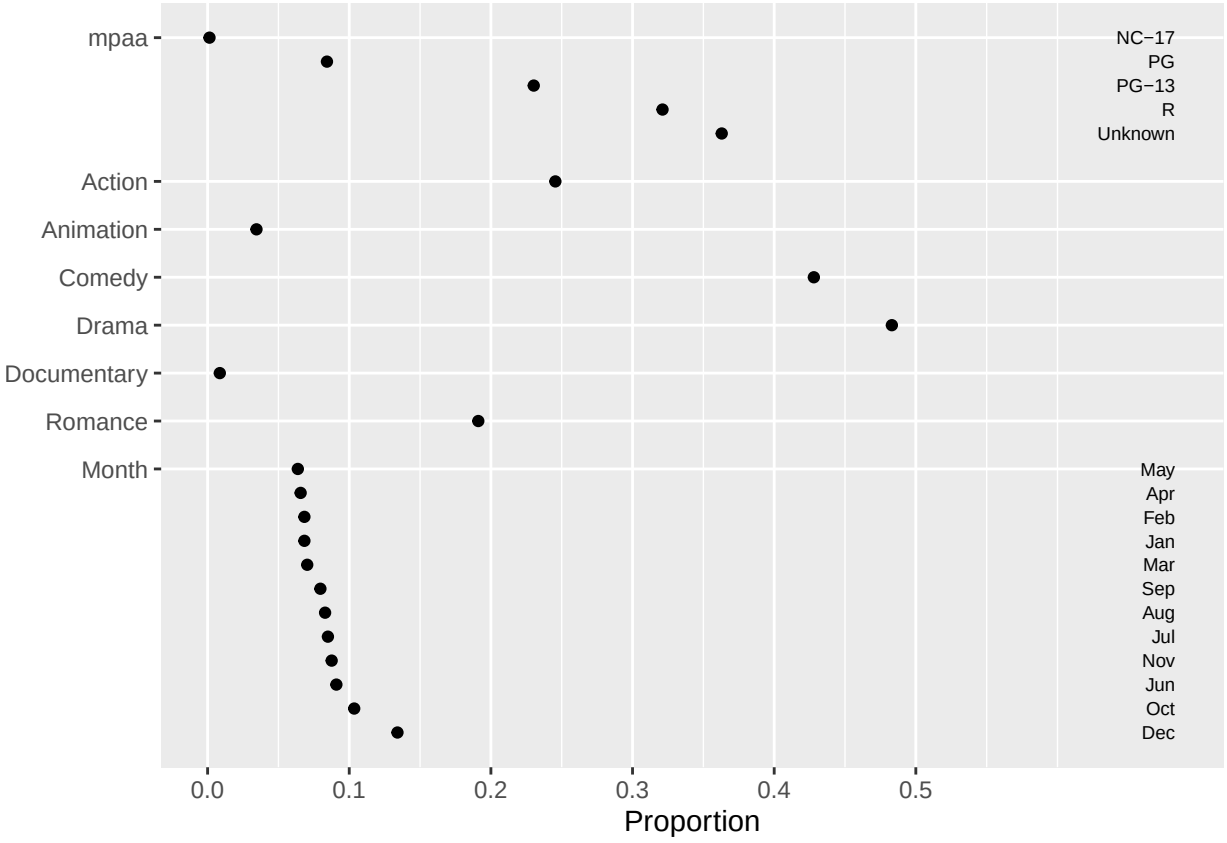
As you can see, `describe` gives us a bit more detailed information than `summary`. See the help file for much more information. Again, not all of it is necessarily useful.

14. The function `describe` also has a nice feature: if you save the results of `describe` into an object, and then `plot` that object, you'll get a couple of plots that might be useful. What do those plots tell you about the distributions of the variables?

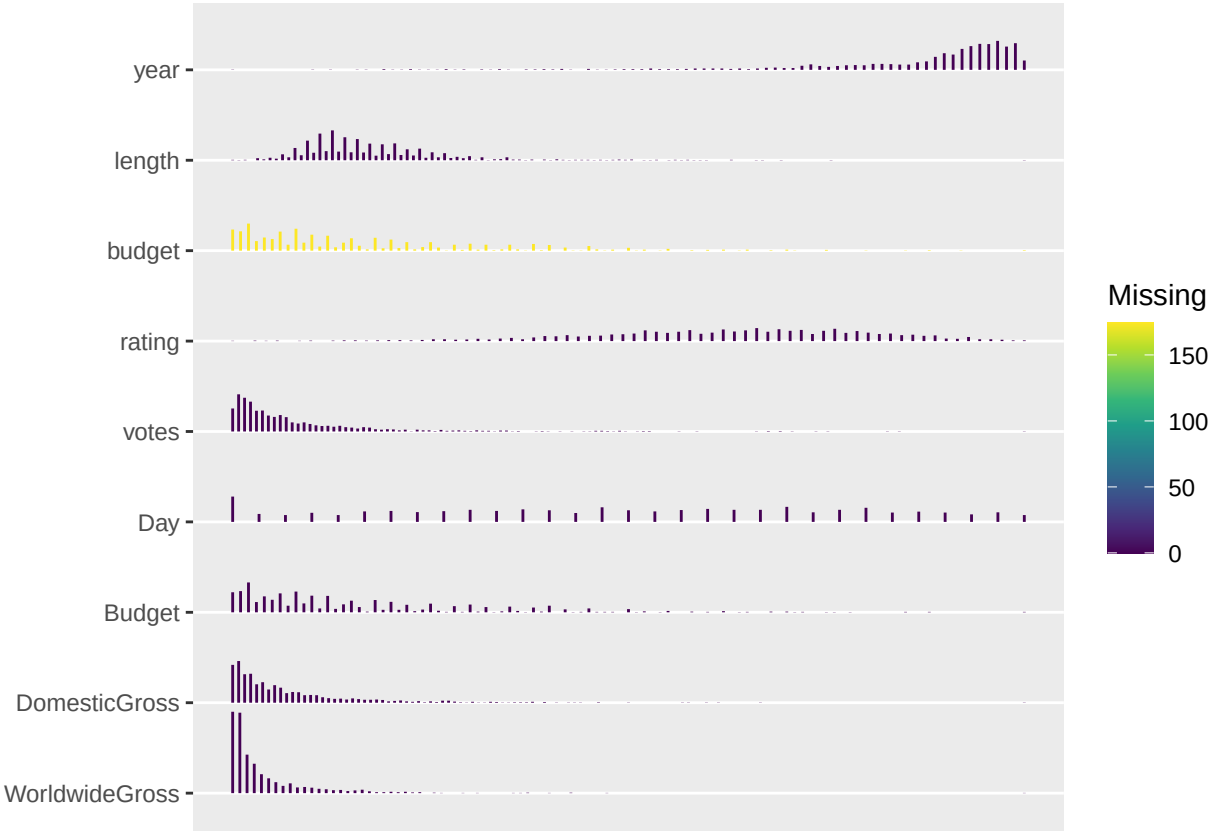
Perhaps more useful than the output of the function `describe` is the output of `plot(describe(dataframe))`. The visual display makes it easier to grasp the distributions of the categorical and quantitative variables. In addition, there is colour-coding to give us an indication of the number of missing values in each variable. Again, see the help file for much more information.

```
DescripData <- describe(AllData)
plot(DescripData)
```

```
$Categorical
```



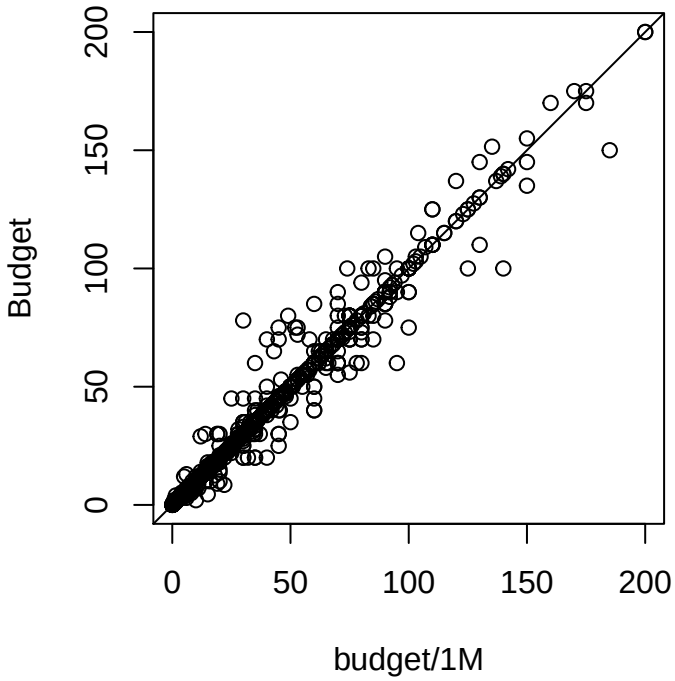

```
$Continuous
```



15. Note that there are two columns with budget information. what are the characteristics of those columns? Do they give the same information? What plot could you construct to determine whether they do?

The variables `Budget` and `budget` (case matters in *R*!) come from the two datasets that we merged in order to form `AllData`. As you can see from the summaries, `budget` gives us the actual dollar amounts, whereas `Budget` is in units of millions of dollars. Furthermore, one has more missing values than the other, and it's of interest to know if they give the same information. There are lots of ways of doing this, but one visual way is to plot the two variables on a scatterplot. If they give exactly the same information, all the points (those that aren't missing) should lie on a straight line.

```
par(pty = "s")  # square plot
plot(Budget ~ I(budget/10^6), data = AllData, xlab = "budget/1M")  # I've divided budget by a million
abline(0, 1)
```



We can clearly see that although many of the films do have the same budget figures, many of them don't!