Intro to R Programming: Class Test 1: MSc 2019

MSc Class Test 1: Movies

The data file `julymovies.csv` contains data on movies which were screened in American cinemas on 30th July since 1999. It contains the following columns:

julymovies.csv	
title	movie title
distributor	company responsible for distributing movie
gross	gross revenue for movie on given date $1000s
theaters	number of theaters screening the movie on the given date
days	number of days the movie had been in theaters by the given date
date	date the row of data corresponds to
top	Indicates if movie is in top grossing 100 American movies of all time(logical)*

Data originally from the-numbers.com, top grossing movies have taken from figures not adjusted for inflation

1. [**2 marks**] Use R to read in the file `julymovies.csv` correctly and save it as a dataframe called `movies`.

```
movies <- read.csv("julymovies.csv", na.strings="*")
```

2. [**2 marks**] Define a variable called `lionsgate` which contains the number of movies within the dataframe `movies` which were distributed by Lionsgate.

```
lionsgate <- sum(movies$distributor=="Lionsgate")
```

3. [**2 marks**] Define a vector called `missing` which contains the number of missing values for each variable in the `movies` dataframe.
   In other words, the vector `missing` should have many elements as there are columns in the `movies` dataframe and each element in `missing` should correspond to one of the columns.

```
missing <- colSums(is.na(movies))
```

4. [**3 marks**] Update the `movies` data frame by removing all rows where the values of `days` and `top` are missing. The updated dataframe should be called `movies`.

```
movies <- subset(movies, !is.na(movies$top)&!is.na(movies$days))
```

5. [**2 marks**] Define a variable called `average.gross` which contains the average revenue per theatre for movies distributed by Walt Disney.

```
disney <- subset(movies, movies$distributor=="Walt Disney")
average.gross <-  sum(disney$gross)/sum(disney$theaters)
```

6. [**2 marks**] Define a variable called `log.gross` which contains, for each movie in the dataframe, the log transformed gross revenue. Add this variable to the `movies` dataframe. (The additional column name should be `log.gross` and the resulting dataframe should still be called `movies`).

```
log.gross <- log(movies$gross)
movies <- transform(movies, log.gross=log.gross)
```

7. [**2 marks**] Sort the `movies` data frame by number of theaters. The sorted dataframe should be called `movies`.

```
movies <- movies[order(movies$theaters),]
```

If you have not managed to complete questions 1 to 7 you can use the following code to generate data which can be used to answer questions 8 to 11.

```
theaters <- c(seq(1,1000,length.out=300), seq(1001,6000,length.out=315))
set.seed(123); log.gross <- log(theaters)+rnorm(615)
```

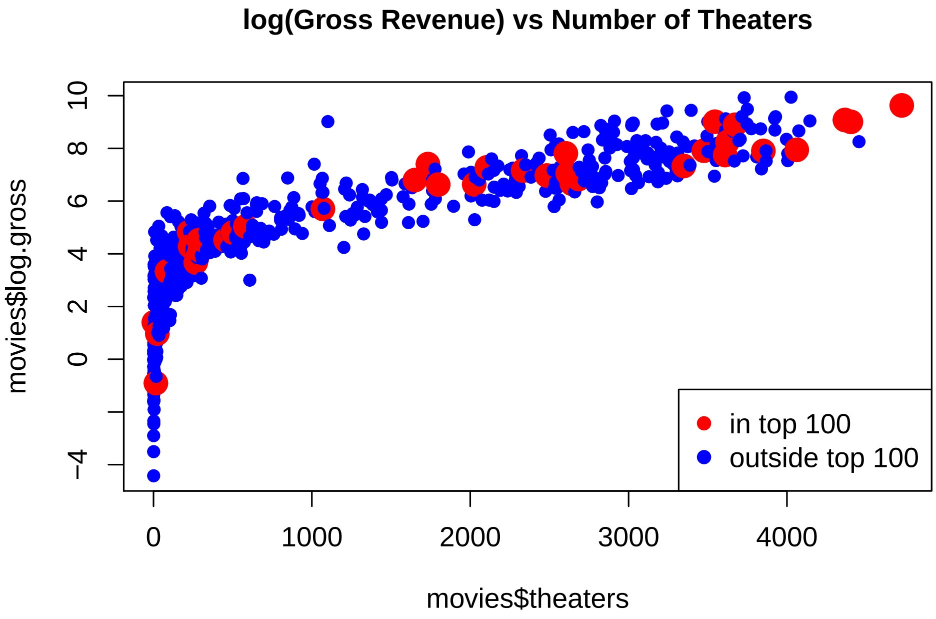
8. [**5 marks**] Produce a plot of `log.gross` by `theaters`.

Your plot should look similar to the one below. On your plot..

- Points representing movies which were in the top 100 grossing American movies of all time should be coloured red, while those which were not in the top 100 should be coloured blue.
- You should set `pch=19` (this changes the plotting character to an solid circle)
- Points representing the movies which were in the top 100 grossing American movies of all time should be twice the size of those representing movies not in the top 100.
- The title of the plot "log(Gross Revenue) vs Number of Theaters".
- There should be a legend the same as the one shown on the plot on page 3.

```
plot(movies$log.gross~ movies$theaters,
     cex.axis=1.1, cex.lab=1.1, cex.main=1.1,
     main="log(Gross Revenue) vs Number of Theaters",
     col=c("blue", "red")[unclass(movies$top)+1],
     cex = unclass(movies$top)+1,
     pch=19)

legend("bottomright", pch=19, legend=c("in top 100", "outside top 100"), col=c(2,4),cex=1.1)
```



9. [**3 marks**] If we want to model the relationship between the number of theaters a movie is shown in (`theaters`) and the log transformed gross revenue a film makes (`log.gross`) we can use fractional polynomial regression.

For a covariate vector $\mathbf{x} = (x_1, ..., x_n)$ the design matrix for fractional polynomial regression of degree 1 takes the form;

$$X = \begin{bmatrix} \frac{1}{x_1} & \frac{1}{\sqrt{x_1}} & 1 & \sqrt{x_1} & x_1 \\ \frac{1}{x_2} & \frac{1}{\sqrt{x_2}} & 1 & \sqrt{x_2} & x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{x_n} & \frac{1}{\sqrt{x_n}} & 1 & \sqrt{x_n} & x_n \end{bmatrix}$$

Define a matrix `X` which is the design matrix required for fitting a fractional polynomial regression model where log transformed gross revenue is the response, $\mathbf{y}$, and number of theaters is the covariate, $\mathbf{x}$.

```r
powers <- c(-1, -0.5, 0, 0.5, 1)
p <- matrix(rep(powers, each=length(movies$theaters)), ncol=5)
x <- matrix(rep(movies$theaters, 5), ncol=5)
X <- x^p

## or
#x <- movies$theaters
#X <- cbind(x^-1, x,^-0.5, x^0, x^0.5, x^1)

## or
## use outer
#X <- outer(movies$theaters, powers, "^")
```

10. [**3 marks**] Using the design matrix, X, from Question 9 define a vector `y.hat` which contains, $\hat{\mathbf{y}} = (\hat{y}_1, ...\hat{y}_n)$, the fitted values for the fractional polynomial regression between the log transformed revenue (response) and number of theaters (covariate).

    The fitted values can be computed using;

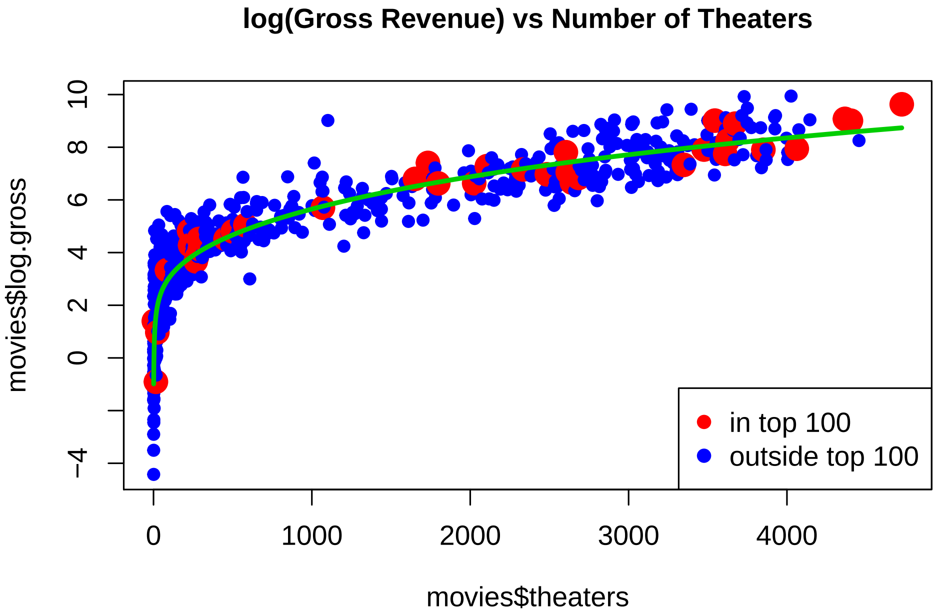
$$\hat{y} = X(X^T X)^{-1} X^T y$$

11. [**2 marks**] Add the fitted fractional polynomial regression line to the plot produced in Question 8. You should add a thick green line to represent the fitted model. Your plot should look like the one below.

```r
plot(movies$log.gross~ movies$theaters,
     cex.axis=1.1, cex.lab=1.1, cex.main=1.1,
     main="log(Gross Revenue) vs Number of Theaters",
     col=c("blue", "red")[unclass(movies$top)+1],
     cex = unclass(movies$top)+1,
     pch=19)

legend("bottomright", pch=19, legend=c("in top 100", "outside top 100"), col=c(2,4),cex=1.1)

lines(movies$theaters, y.hat, col=3, lwd=3)
```



12. [**4 marks**] R^2 can be used as a measure of how well a regression line fits a set of data. For a response variable denoted $\mathbf{y} = y_1, ...y_n$, R^2 can be calculated as

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2}$$

where $\hat{\mathbf{y}} = (\hat{y}_1, ..\hat{y}_n)$are the fitted values and $\bar{y} = \frac{\sum_i^n y_i}{n}$ is the mean of y.

Define a variable `Rsq` which contains the R^2 value for the fitted fractional polynomial regression line computed in Question 10.

```
y <- log.gross
n <- length(y)
f <- y.hat
Rsq <-  1-(sum((y-f)^2)/sum((y-mean(y))^2))
```

13. [**4 marks**] Susan wants to see a movie which starts at exactly 8pm. She leaves her house to go to the cinema randomly at any time between 7.00pm and 7.30pm and her journey there can take anywhere between 30 and 45 minutes depending on traffic. Assume that the length of her journey is also random.

Use a simulation based on 1000 possible journeys to define a vector called `late` which contains the probability Susan arrives after the movie starts at 8pm.

Please enter and run the line of code below before carrying out your simulation.

```
set.seed(123)
```

Hint: You can use the function `runif(n,a,b)` to generate `n` draws from a random uniform distribution with lower limit `a` and upper limit `b`.

```
leaves <- runif(1000,0,30)
journey <- runif(1000, 30,45)
total <- leaves +journey
late <- sum(total >= 60)/1000
```

14. [**4 marks**] For this question you should answer based on your simulation from Question 13.

i) Define a variable called `waiting` which contains the average number of minutes that Susan has to wait before the film begins.

ii) Define a variable called `unseen` which contains average number of minutes of the movie Susan will miss if she is not there for the start of the movie.

```r
waiting <- abs(mean(total[total<60])-60)
unseen <- mean(total[total>60])-60
```