HW 5 (116 points)

STAT 131A Fall 2021

Due Date: December 8, 2021

Question 1

a (2 points) I have a dataset containing average hourly earnings in dollars (wage) and years of education (educ) for 526 individuals. I fit a simple linear regression equation with wage as response and educ as the explanatory variable. This gave me the following equation:

wage = -0.90485 + 0.54136 * (educ).

Which among the following is the correct interpretation for this equation? Give reasons for your answer.

(i) For every additional four years of education, the average hourly wage increases by 4*0.54 = 2.16 dollars.

(ii) For every additional year of education, the average hourly wage increases by 54%.

(iii) For every 1% increase in the number of years of education, the average hourly wage increases by 0.54%.

> My answer:
> (i) is correct, since the scope of 0.54 represent that for one unit increase in 'educ', wage will go up in 0.54 unit. If we increase years of education by 4, the wage will go up by 4*0.54.
> And (ii) and (iii) are wrong because the scope does not convert the unit increase in 'educ' into the increased percentage in wage.

b (2 points) For the same dataset as in the previous part, I fit a simple linear regression equation with log(wage) as response and educ as the explanatory variable. This gave me the following equation:

log(wage) = 0.583773 + 0.082744 * (educ).

Which among the following is the correct interpretation for this equation? Give reasons for your answer.

(i) For every additional year of education, the average hourly wage increases by 0.0827 dollars.

(ii) For every additional year of education, the average hourly wage increases by 8.27 percent.

(iii) For every additional year of education, the average hourly wage increases by 0.0827 percent.

> My answer:
> (ii) is correct. Using $log(1+t) \approx t$, we can get that the increase in 'log(wage)' approximately equals to the increased percentage of 'wage', which equals to '0.082744 * (educ)'. That is to say, for every additional year of education, the average 'log(wage)' will increase 0.0827 unit and the average wage will go up by 8.27%.

c (2 points) I have a dataset on the salaries of the CEOs of 209 firms (variable name is salary) along with the sales of the firm (variable is sales). The dataset is from the year 1990. Salary is in thousands of dollars and Sales is in millions of dollars. I fit a simple linear regression with log(salary) as the response variable and log(sales) as the explanatory variable and this gave me the equation:

log(salary) = 4.822 + 0.25667 * log(sales).

Which among the following is the correct interpretation for this equation? Give reasons for your answer.

(i) For a 1 percent increase in sales, the CEO salary increases by 0.257 percent on average.

(ii) For a 1 million dollar increase in firm sales, the CEO salary increases by 25.667 thousand dollars on average.

(iii) For a 1 million dollar increase in firm sales, the CEO salary increases by 2.57 percent.

My answer:
(i) is correct. Using $log(1 + t) \approx t$, we can get that the increased percentage of 'sales' approximately equals to the unit increase in 'log(sales)', and the unit increase in 'log(wage)' approximately equals to the increased percentage of 'wage'. That is to say, for a 1% increase in sales, 'log(sales)' will increase by 0.01 unit, then the average 'log(salary)' will increase by 0.00257 unit and the average wage will increase by 0.257%.

Question 2

(15 points) The following is a the output of running `lm` on a subset of the imdb dataset you will work with below (Question 4). The below output above has five missing values which are indicated by XXAXX-XXFXX Using only the available information in the above summary, fill in the missing values. I give you space below for R code, but this is just for using R as a calculator – you can't recreate this lm summary with the data given, because this is done on a random subset of the full dataset.

```
XXAXX = (1.079e-05 - 0)/ 1.095e-05
XXAXX
```

```
## [1] 0.9853881
```

```
XXBXX = 2 * (1 - pt(0.9853881, df = 558))
XXBXX
```

```
## [1] 0.3248605
```

```
XXFXX = pf(34.16, 13, 558, lower.tail = F)
XXFXX
```

```
## [1] 1.397271e-62
```

My answer:
XXAXX = 0.9853881 XXBXX = 0.3248605 XXCXX = 13 XXDXX = 558
XXFXX $=1.3e-62 \approx 0$

```
summary(lmMoviesSmall2)

Call:
lm(formula = imdb_score ~ ., data = moviesSmall2)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6138 -0.4630  0.0876  0.5490  1.9408

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  4.425e+01  9.482e+00   4.667 3.84e-06 ***
```

```
num_critic_for_reviews     2.540e-03  5.055e-04   5.025 6.78e-07 ***
duration                   1.111e-02  1.710e-03   6.497 1.81e-10 ***
director_facebook_likes    1.079e-05  1.095e-05   XXAXX    XXBXX
actor_3_facebook_likes     9.128e-05  5.226e-05   1.747  0.08126 .
actor_1_facebook_likes     8.848e-05  3.153e-05   2.807  0.00518 **
gross                      1.662e-10  6.693e-10   0.248  0.80399
num_voted_users            3.746e-06  4.309e-07   8.694  < 2e-16 ***
cast_total_facebook_likes -7.583e-05  3.127e-05  -2.425  0.01564 *
num_user_for_reviews      -7.565e-04  1.575e-04  -4.804 2.01e-06 ***
budget                    -4.223e-09  1.025e-09  -4.122 4.33e-05 ***
title_year                -1.973e-02  4.727e-03  -4.175 3.46e-05 ***
actor_2_facebook_likes     6.026e-05  3.242e-05   1.859  0.06362 .
movie_facebook_likes      -1.150e-06  2.366e-06  -0.486  0.62723
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8023 on 558 degrees of freedom
Multiple R-squared:  0.4432,    Adjusted R-squared:  0.4302
F-statistic: 34.16 on XXCXX and XXDXX DF,  p-value: XXFXX
```

Question 3

Consider the data in `ceodata_num.csv` which consists of 209 firms and has data on the salary of the CEO, sales of the firm, and the firm type. The data is from the year 1990.

```r
ceodata<-read.csv("ceodata_num.csv")
```

`salary`: Salary of the CEO in thousands of dollars `sales`: Sales of company in millions of dollars `FirmType`: the type of company as numeric values 1-4 which correspond to:

- 1=consumer product
- 2=finance
- 3=industry
- 4=utility

a (10 points) Fit a regression in R with `sales` and `FirmType` as a predictor of `salary`. Interpret each of the coefficient estimates given by R, except for the intercept.

```r
# Code for fitting regression here
ceodata$FirmType<-factor(ceodata$FirmType,levels=c(1,2,3,4),labels=c("consumer product ","finance","indu
fit = lm(salary~sales+FirmType, data = ceodata)
summary(fit)
```

```
##
## Call:
## lm(formula = salary ~ sales + FirmType, data = ceodata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1620.3  -425.8  -162.1    71.2 13173.5
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.621e+03  1.866e+02   8.689  1.2e-15 ***
## sales             1.249e-02  8.833e-03   1.414  0.15882
## FirmTypefinance  -3.496e+02  2.625e+02  -1.332  0.18444
## FirmTypeindustry -5.867e+02  2.374e+02  -2.471  0.01429 *
```

```
## FirmTypeutility  -9.396e+02  2.842e+02  -3.305  0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1336 on 204 degrees of freedom
## Multiple R-squared:  0.07097,    Adjusted R-squared:  0.05275
## F-statistic: 3.896 on 4 and 204 DF,  p-value: 0.004517
```

> My answer:
> The baseline model is 'salary = 1.249e-02 * sales + 1.621e+03',when 'FirmType = consumer product'. The slope for sales means that, holding everything else constant, one unit increase in sales will lead to 1.249e-02 unit increase in salary on average.
> When 'FirmType = finance', 'salary = 1.249e-02 * sales + 1.621e+03 - 3.496e+02'. 'FirmTypefinance' means that, there will be 3.496e+02 decrease in the average salary relative to the consumer product firm.
> When 'FirmType = industry', 'salary = 1.249e-02 * sales + 1.621e+03 - 5.867e+02'. 'FirmTypeindustry' means that, there will be 5.867e+02 decrease in the average salary relative to the consumer product firm.
> When 'FirmType = utility', 'salary = 1.249e-02 * sales + 1.621e+03 - 9.396e+02'. 'FirmTypeutility' means that, there will be 9.396e+02 decrease in the average salary relative to the consumer product firm.

b (10 points) Fit a regression that allows for a different slope for the different types of firms. Give a summary of the results and interpret each coefficient, except for the intercept.

```r
# Code for different slopes for each type of firm
fit2 = lm(salary~sales+FirmType+sales:FirmType, data = ceodata)
summary(fit2)
```

```
##
## Call:
## lm(formula = salary ~ sales + FirmType + sales:FirmType, data = ceodata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1513.0  -430.5  -143.2    68.9 13089.4
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.736e+03  1.991e+02   8.719 1.06e-15 ***
## sales                  -1.715e-03  1.235e-02  -0.139  0.88967
## FirmTypefinance        -6.164e+02  3.586e+02  -1.719  0.08716 .
## FirmTypeindustry       -8.060e+02  2.810e+02  -2.868  0.00457 **
## FirmTypeutility        -1.375e+03  5.179e+02  -2.656  0.00855 **
## sales:FirmTypefinance   4.023e-02  4.028e-02   0.999  0.31914
## sales:FirmTypeindustry  2.669e-02  1.827e-02   1.461  0.14557
## sales:FirmTypeutility   1.013e-01  1.155e-01   0.877  0.38136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1335 on 201 degrees of freedom
## Multiple R-squared:  0.08562,    Adjusted R-squared:  0.05377
## F-statistic: 2.689 on 7 and 201 DF,  p-value: 0.01105
```

My answer is:
The baseline model is 'salary = -1.715e-03 * sales + 1.736e+03',when 'FirmType = consumer product'. The slope for sales means that, holding everything else constant, one unit increase in sales will lead to 1.715e-03 unit decrease in salary on average.
When 'FirmType = finance', 'salary = (-1.715e-03 + 4.023e-02) * sales + 1.736e+03 - 6.164e+02'. 'FirmTypefinance' means that, holding everything else constant, one unit increase in sales will lead to 4.023e-02 increase in slope and 6.164e+02 decrease in intercept.
When 'FirmType = industry', 'salary = (-1.715e-03 + 2.669e-02) * sales + 1.736e+03 - 8.060e+02'. 'FirmTypeindustry' means that, holding everything else constant, one unit increase in sales will lead to 2.669e-02 increase in slope and 8.060e+02 decrease in intercept.
When 'FirmType = utility', 'salary = (-1.715e-03 + 1.013e-01) * sales + 1.736e+03 - 1.375e+03'. 'FirmTypeutility' means that, holding everything else constant, one unit increase in sales will lead to 1.013e-01 increase in slope and 1.375e+03 decrease in intercept.

c. (10 points) Evaluate whether this model in part b is an improvement over the model in part a.

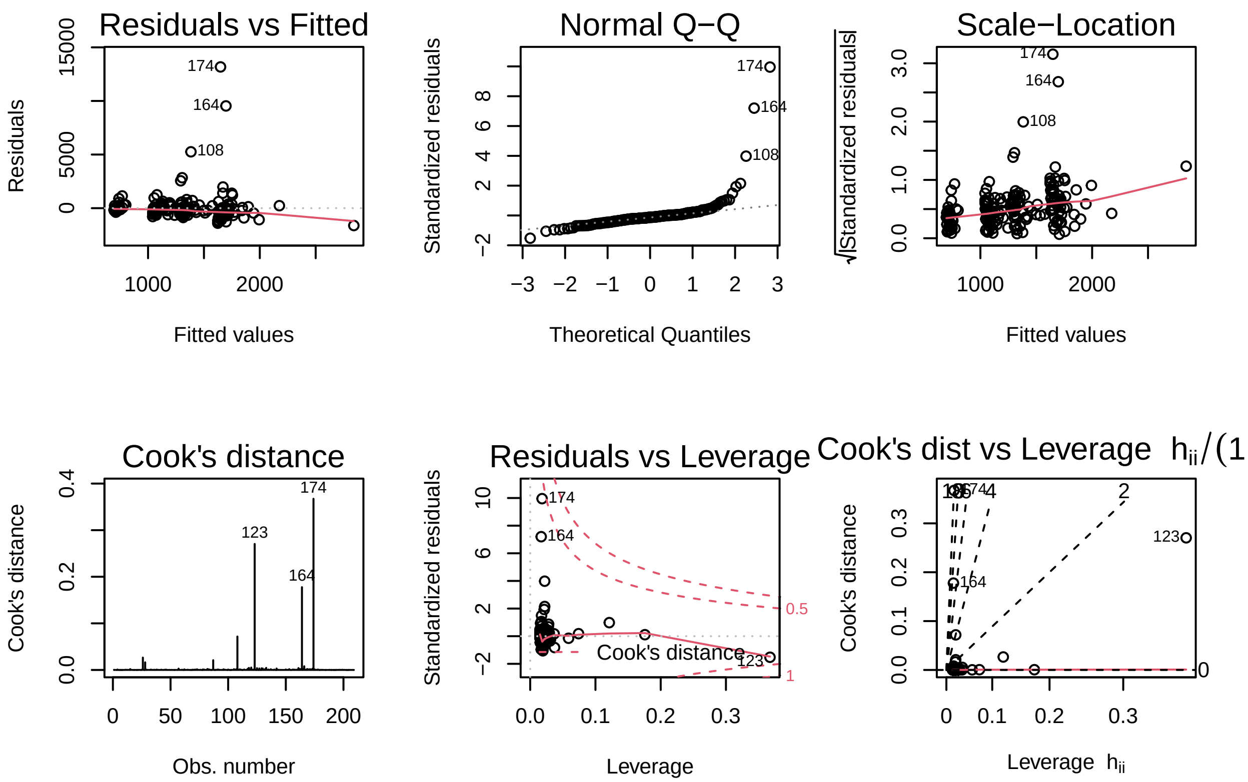
```r
# Code for evaluating model
LOOCV<-function(lm){
    vals<-residuals(lm)/(1-lm.influence(lm)$hat)
    sum(vals^2)/length(vals)
}
data.frame(
  AIC = c(AIC(fit),AIC(fit2)),
  LOOCV = c(LOOCV(fit),LOOCV(fit2)),
  row.names = c("model a","model b")
)
```

```
##              AIC   LOOCV
## model a 3608.476 1828020
## model b 3611.155 1801133
```

My answer is:
Based on AIC, model in part b is not an improvement over the model in part a, since model b has a higher AIC value.
However, based on LOOCV, model in part b is an improvement over the model in part a, since model b has a lower LOOCV value.

d. (20 points) Run diagnostics on the model you found in part a, and determine whether there are any problems with this model that should be addressed. If so, explain next steps you might take to try to improve this model.

```r
# Code for diagnostics
par(mfrow = c(2, 3))
plot(lm(salary~sales+FirmType, data=ceodata), which=1:6)
```



My answer:
There are any problems with this model.
We can see there's a linear relation in the Residuals vs Fitted plot, and increasing pattern the Scale-Location plot, suggesting that the distribution of residuals is heteroscedastic.
From the QQ plot, we can see that residuals are not normally distributed, suggesting the distribution is light-tailed.
From the Cook's distance plot and Residuals vs Leverage plot, we can see that there are some outliers such as 174, 164 and 123.
The next steps can be: (1)remove outliers; (2) add in more explanatory variables since there's a remaining linear relation in the residuals plot; (3) do variable selection to fine the best model.

Question 4

Consider data from the website www.imdb.com giving scores of movies. Read in the data with the following code:

```r
movies<-read.csv("imdb_simplified.csv",header=TRUE)
movies<-movies[,-grep("name",names(movies))]   ## LINE2
movies<-movies[,!names(movies) %in% "movie_title"] ## LINE3
head(movies)
```

```
##   num_critic_for_reviews duration director_facebook_likes
## 1                    723      178                       0
## 2                    302      169                     563
## 3                    813      164                   22000
```

```
## 4                      462      132                     475
## 5                      324      100                      15
## 6                      635      141                       0
##   actor_3_facebook_likes actor_1_facebook_likes     gross num_voted_users
## 1                    855                   1000 760505847          886204
## 2                   1000                  40000 309404152          471220
## 3                  23000                  27000 448130642         1144337
## 4                    530                    640  73058679          212204
## 5                    284                    799 200807262          294810
## 6                  19000                  26000 458991599          462669
##   cast_total_facebook_likes num_user_for_reviews content_rating    budget
## 1                      4834                 3054          PG-13 237000000
## 2                     48350                 1238          PG-13 300000000
## 3                    106759                 2701          PG-13 250000000
## 4                      1873                  738          PG-13 263700000
## 5                      2036                  387             PG 260000000
## 6                     92000                 1117          PG-13 250000000
##   title_year actor_2_facebook_likes imdb_score movie_facebook_likes
## 1       2009                    936        7.9                33000
## 2       2007                   5000        7.1                    0
## 3       2012                  23000        8.5               164000
## 4       2012                    632        6.6                24000
## 5       2010                    553        7.8                29000
## 6       2015                  21000        7.5               118000
##                genres
## 1              Action
## 2              Action
## 3     Thriller_Action
## 4              Action
## 5 Romance_Comedy_Action
## 6              Action
```

a. (5 points) In the code above, explain what LINE 2 and LINE3 are doing (see `?grep`).

> My answer:
> LINE2 remove columns whose column names contains "name" substring LINE3 remove the 'movie_title' column, since "movie_title" string is in 'names(movies)', making '!names(movies) %in% "movie_title" ' condition 'FALSE'.

b. (10 points) Find best submodel based on AIC,*without including the categorical variables (`genre` or `content_rating`)*. (Hint: If you use `regsubsets`, that function requires you to set `nvmax` as the maximum size submodel you want to consider, and thus have to set it larger than the largest possible model if you want to compare all possible sizes)

```
# The best submodel based on AIC
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.1.2
```

```
movies.clean = movies[, -c(10,16)]
bMovie = regsubsets(imdb_score ~ ., movies.clean, nvmax = 14)

LOOCV<-function(lm){
    vals<-residuals(lm)/(1-lm.influence(lm)$hat)
    sum(vals^2)/length(vals)
```

```r
}
calculateCriterion<-function(x=NULL,y,dataset,lmObj=NULL){
    #dataset contains only explanatory variables
    #x is a vector of logicals, length equal to number of explanatory variables in dataset, telling us w
    #sigma2 is estimate of model on full dataset
    # either x or lmObj must be given to specify the smaller lm model
    sigma2=summary(lm(y~.,data=dataset))$sigma^2
    if(is.null(lmObj)) lmObj<-lm(y ~ ., data=dataset[,x,drop=FALSE]) #don't include intercept
    sumlmObj<-summary(lmObj)
    n<-nrow(dataset)
    p<-sum(x)
    RSS<-sumlmObj$sigma^2*(n-p-1)
    c(R2=sumlmObj$r.squared,
        R2adj=sumlmObj$adj.r.squared,
        "RSS/n"=RSS/n,
        LOOCV=LOOCV(lmObj),
        Cp=RSS/n+2*sigma2*(p+1)/n,
        CpAlt=RSS/sigma2-n+2*(p+1),
        AIC=AIC(lmObj), # n*log(RSS/n)+2*p +constant,
        BIC=BIC(lmObj) # n*log(RSS/n)+p*log(n) + constant
    )
}
critMovie<-apply(summary(bMovie)$which[,-1],1,calculateCriterion,
    y=movies.clean$imdb_score,
    dataset=movies.clean[,-13])
critMovie<-t(critMovie)
critMovie[,7]
```

```
##        1        2        3        4        5        6        7        8
## 7096.405 6901.297 6775.682 6713.787 6619.242 6535.713 6527.589 6517.069
##        9       10       11       12       13
## 6506.918 6504.293 6503.695 6503.849 6505.083
```

```r
data.frame(
  AIC = which.min(abs(critMovie[,"AIC"])),
  LOOCV = which.min(abs(critMovie[,"LOOCV"]))
)
```

```
##    AIC LOOCV
## 11  11    11
```

```r
summary(bMovie)$out
```

```
##           num_critic_for_reviews duration director_facebook_likes
## 1  ( 1 )  " "                    " "      " "
## 2  ( 1 )  " "                    "*"      " "
## 3  ( 1 )  " "                    "*"      " "
## 4  ( 1 )  "*"                    "*"      " "
## 5  ( 1 )  "*"                    "*"      " "
## 6  ( 1 )  "*"                    "*"      " "
## 7  ( 1 )  "*"                    "*"      " "
## 8  ( 1 )  "*"                    "*"      " "
## 9  ( 1 )  "*"                    "*"      " "
## 10  ( 1 ) "*"                    "*"      " "
## 11  ( 1 ) "*"                    "*"      " "
```

```
## 12  ( 1 ) "*"                    "*"      " "
## 13  ( 1 ) "*"                    "*"      "*"
##           actor_3_facebook_likes actor_1_facebook_likes gross num_voted_users
## 1  ( 1 )  " "                    " "                    " "   "*"
## 2  ( 1 )  " "                    " "                    " "   "*"
## 3  ( 1 )  " "                    " "                    " "   "*"
## 4  ( 1 )  " "                    " "                    " "   "*"
## 5  ( 1 )  " "                    " "                    " "   "*"
## 6  ( 1 )  " "                    " "                    " "   "*"
## 7  ( 1 )  "*"                    " "                    " "   "*"
## 8  ( 1 )  "*"                    "*"                    " "   "*"
## 9  ( 1 )  " "                    "*"                    " "   "*"
## 10  ( 1 ) " "                    "*"                    " "   "*"
## 11  ( 1 ) "*"                    "*"                    " "   "*"
## 12  ( 1 ) "*"                    "*"                    "*"   "*"
## 13  ( 1 ) "*"                    "*"                    "*"   "*"
##           cast_total_facebook_likes num_user_for_reviews budget title_year
## 1  ( 1 )  " "                       " "                  " "    " "
## 2  ( 1 )  " "                       " "                  " "    " "
## 3  ( 1 )  " "                       " "                  "*"    " "
## 4  ( 1 )  " "                       " "                  "*"    " "
## 5  ( 1 )  " "                       " "                  "*"    "*"
## 6  ( 1 )  " "                       "*"                  "*"    "*"
## 7  ( 1 )  " "                       "*"                  "*"    "*"
## 8  ( 1 )  " "                       "*"                  "*"    "*"
## 9  ( 1 )  "*"                       "*"                  "*"    "*"
## 10  ( 1 ) "*"                       "*"                  "*"    "*"
## 11  ( 1 ) "*"                       "*"                  "*"    "*"
## 12  ( 1 ) "*"                       "*"                  "*"    "*"
## 13  ( 1 ) "*"                       "*"                  "*"    "*"
##           actor_2_facebook_likes movie_facebook_likes
## 1  ( 1 )  " "                    " "
## 2  ( 1 )  " "                    " "
## 3  ( 1 )  " "                    " "
## 4  ( 1 )  " "                    " "
## 5  ( 1 )  " "                    " "
## 6  ( 1 )  " "                    " "
## 7  ( 1 )  " "                    " "
## 8  ( 1 )  " "                    " "
## 9  ( 1 )  "*"                    " "
## 10  ( 1 ) "*"                    "*"
## 11  ( 1 ) "*"                    "*"
## 12  ( 1 ) "*"                    "*"
## 13  ( 1 ) "*"                    "*"
```

The best model is the 11th model with size 11, that is 'lm(imdb__score~num__critic__for__reviews+duration+actor__3__facebook__likes +actor__1__facebook__likes+num__voted__users+ cast__total__facebook__likes +num__user__for__reviews budget)',with the lowest AIC of 6503.695.

c. (10 points) Use CV to compare the best models of each possible size K, as found by comparing RSS.

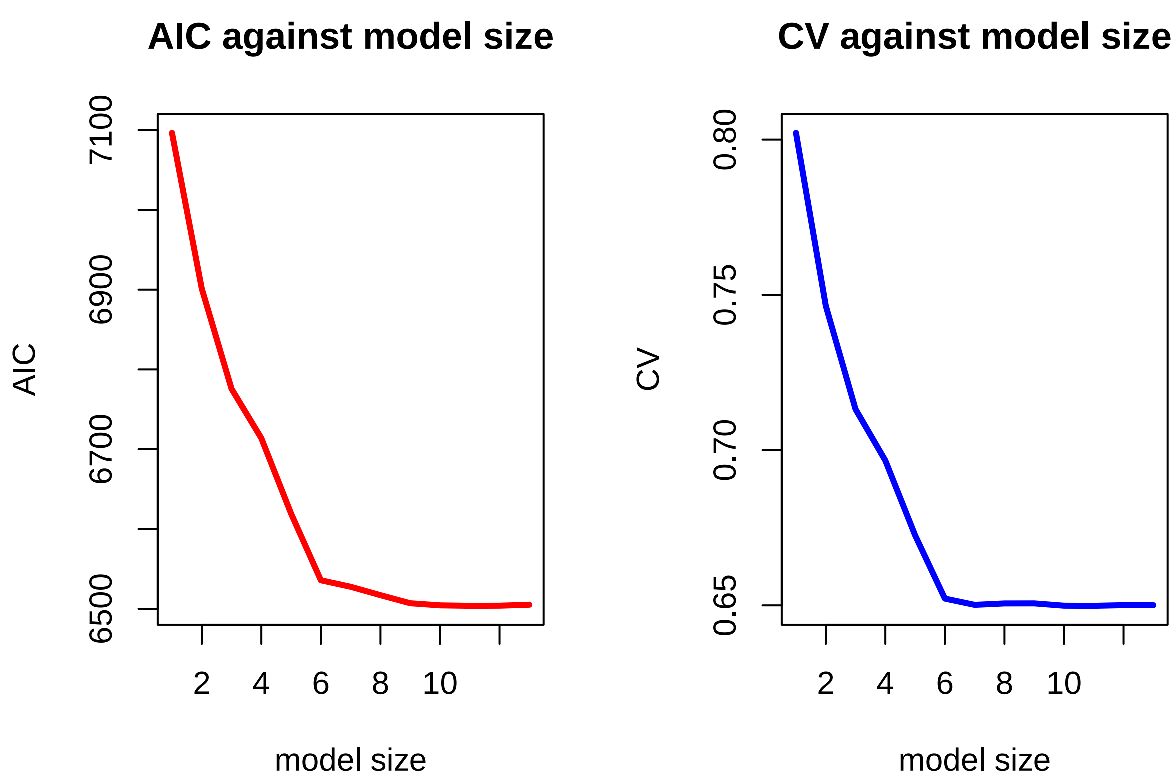
```r
# The best submodel based on CV
critMovie[,4]
```

```
##         1         2         3         4         5         6         7         8 
## 0.8021437 0.7464838 0.7131653 0.6966520 0.6725623 0.6521780 0.6501586 0.6506159 
##         9        10        11        12        13 
## 0.6506353 0.6498823 0.6498450 0.6500645 0.6500538
```

> The best model is the 11th model with size 11, that is ‘lm(imdb_score~num_critic_for_reviews+duration+actor_3_facebook_likes+actor_1_facebook_likes+num_voted_ cast_total_facebook_likes+num_user_for_reviews budget)’,with the lowest LOOCV of 0.6498450.

d. (5 points) Plot the AIC and CV as a function of model size, and comment on whether AIC and CV lead to the same answer.

```r
# plots of aic and cv vs. model size
AIC = critMovie[,7]
CV = critMovie[,4]
par(mfrow= c(1,2))
plot(c(1:13), AIC, lwd = 3, col = "red", type = "l", main = "AIC against model size",
     xlab = "model size",ylab = "AIC")
plot(c(1:13), CV, lwd = 3, col = "blue", type = "l", main = "CV against model size",
     xlab = "model size",ylab = "CV")
```



My answer:
AIC and CV are quite similar and lead to the same answer, which are both highest at the null model and drop as we add in more predictors, reaching the optimal model at 11.

e. (10 points) If you had far more variables, you would not be able to find the best among all submodels, and could use stepwise regression. Use the `step` function to find a good submodel (based on AIC). Does it find best model based on AIC? If not, report the AIC of the model `step` does find.

```
# applying step
stepwise.model = step(lm(imdb_score~.,data = movies.clean),trace=0, direction="both")
stepwise.model
```

```
##
## Call:
## lm(formula = imdb_score ~ num_critic_for_reviews + duration +
##     actor_3_facebook_likes + actor_1_facebook_likes + num_voted_users +
##     cast_total_facebook_likes + num_user_for_reviews + budget +
##     title_year + actor_2_facebook_likes + movie_facebook_likes,
##     data = movies.clean)
##
## Coefficients:
##               (Intercept)     num_critic_for_reviews
##                 4.955e+01                  2.932e-03
##                  duration     actor_3_facebook_likes
##                 1.210e-02                  3.435e-05
##    actor_1_facebook_likes            num_voted_users
##                 4.852e-05                  3.496e-06
## cast_total_facebook_likes       num_user_for_reviews
##                -4.569e-05                 -6.295e-04
##                    budget                 title_year
##                -4.659e-09                 -2.242e-02
##    actor_2_facebook_likes       movie_facebook_likes
##                 5.137e-05                 -2.251e-06
```

```
AIC(stepwise.model)
```

```
## [1] 6503.695
```

My answer:
the step function finds the best model based on AIC, which is 'lm(imdb_score~num_critic_for_reviews+duration+actor_3_facebook_likes+actor_1_facebook_likes+num_voted_
cast_total_facebook_likes+num_user_for_reviews budget)', the same as part a and b, with the AIC of 6503.695

f. (5 points) In the help of `regsubsets` it states that it will not run if there are more than 50 variables (because it will take too long to try all of the submodels).

If I try to run `regsubsets` on the 15 explanatory variables in the dataset `movie`, i.e. include the categorical variables `genre` and `content_rating`, it says that it has reached that limit:

```
> regsubsets(imdb_score~., movies,nvmax=30)
Error in leaps.exhaustive(a, really.big) :
  Exhaustive search will be S L O W, must specify really.big=T
```

Given an explanation for how `regsubsets` determined there are more than 50 variables.

My answer:
Since the categorical variables `genre` and `content_rating` are not excluded, `regsubsets` function takes all unique values of categorical variables as predictors and there are more than 50 variables (including the original 15 explanatory variables and different values in `genre` and `content_rating`).