Linear_Regression1

author: Dr. Marko Mitic

Problem Description: **READING TEST SCORES**

The Programme for International Student Assessment (PISA) is a test given every three years to 15-year-old students from around the world to evaluate their performance in mathematics, reading, and science. This test provides a quantitative way to compare the performance of students from different parts of the world. In here, we will predict the reading scores of students from the United States of America on the 2009 PISA exam.

The datasets pisa2009train.csv and pisa2009test.csv contain information about the demographics and schools for American students taking the exam, derived from 2009 PISA Public-Use Data Files (http://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2011038), distributed by the United States National Center for Education Statistics (NCES). While the datasets are not supposed to contain identifying information about students taking the test, by using the data you are bound by the NCES data use agreement, which prohibits any attempt to determine the identity of any student in the datasets.

Each row in the datasets pisa2009train.csv and pisa2009test.csv represents one student taking the exam. The datasets have the following variables:

-**grade**: The grade in school of the student (most 15-yearolds in America are in 10th grade)

-**male**: Whether the student is male (1/0)

-**raceeth**: The race/ethnicity composite of the student

-**preschool**: Whether the student attended preschool (1/0)

-**expectBachelors**: Whether the student expects to obtain a bachelor's degree (1/0)

-**motherHS**: Whether the student's mother completed high school (1/0)

-**motherBachelors**: Whether the student's mother obtained a bachelor's degree (1/0)

-**motherWork**: Whether the student's mother has part-time or full-time work (1/0)

-**fatherHS**: Whether the student's father completed high school (1/0)

-**fatherBachelors**: Whether the student's father obtained a bachelor's degree (1/0)

-**fatherWork**: Whether the student's father has part-time or full-time work (1/0)

-**selfBornUS**: Whether the student was born in the United States of America (1/0)

-**motherBornUS**: Whether the student's mother was born in the United States of America (1/0)

-**fatherBornUS**: Whether the student's father was born in the United States of America (1/0)

-**englishAtHome**: Whether the student speaks English at home (1/0)

-**computerForSchoolwork**: Whether the student has access to a computer for schoolwork (1/0)

-**read30MinsADay**: Whether the student reads for pleasure for 30 minutes/day (1/0)

-**minutesPerWeekEnglish**: The number of minutes per week the student spend in English class

-**studentsInEnglish**: The number of students in this student's English class at school

-**schoolHasLibrary**: Whether this student's school has a library (1/0)

-**publicSchool**: Whether this student attends a public school (1/0)

-**urban**: Whether this student's school is in an urban area (1/0)

-**schoolSize**: The number of students in this student's school

-readingScore: The student's reading score, on a 1000-point scale

By looking and the structure of the dataset, it is ovious that 3663 students are investigated in the training dataset. It also can be observed that many of the variables contain NA values.

```
pisa2009train=read.csv("pisa2009train.csv")
pisa2009test=read.csv("pisa2009test.csv")

str(pisa2009train)
```

```
## 'data.frame':    3663 obs. of  24 variables:
##  $ grade                : int  11 11 9 10 10 10 10 10 9 10 ...
##  $ male                 : int  1 1 1 0 1 1 0 0 0 1 ...
##  $ raceeth              : Factor w/ 7 levels "American Indian/Alaska Native",..: NA 7 7 3 4 3 2 7 7 5
##  $ preschool            : int  NA 0 1 1 1 1 0 1 1 1 ...
##  $ expectBachelors      : int  0 0 1 1 0 1 1 1 0 1 ...
##  $ motherHS             : int  NA 1 1 0 1 NA 1 1 1 1 ...
##  $ motherBachelors      : int  NA 1 1 0 0 NA 0 0 NA 1 ...
##  $ motherWork           : int  1 1 1 1 1 1 1 0 1 1 ...
##  $ fatherHS             : int  NA 1 1 1 1 1 NA 1 0 0 ...
##  $ fatherBachelors      : int  NA 0 NA 0 0 0 NA 0 NA 0 ...
##  $ fatherWork           : int  1 1 1 1 0 1 NA 1 1 1 ...
##  $ selfBornUS           : int  1 1 1 1 1 1 0 1 1 1 ...
##  $ motherBornUS         : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ fatherBornUS         : int  0 1 1 1 0 1 NA 1 1 1 ...
##  $ englishAtHome        : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ computerForSchoolwork: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ read30MinsADay       : int  0 1 0 1 1 0 0 1 0 0 ...
##  $ minutesPerWeekEnglish: int  225 450 250 200 250 300 250 300 378 294 ...
##  $ studentsInEnglish    : int  NA 25 28 23 35 20 28 30 20 24 ...
##  $ schoolHasLibrary     : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ publicSchool         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ urban                : int  1 0 0 1 1 0 1 0 1 0 ...
##  $ schoolSize           : int  673 1173 1233 2640 1095 227 2080 1913 502 899 ...
##  $ readingScore         : num  476 575 555 458 614 ...
```

Next, using `tapply` function we can see the result of (for example) reading score by gender:

```
tapply(pisa2009train$readingScore, pisa2009train$male, mean, na.rm=TRUE)
```

```
##        0        1
## 512.9406 483.5325
```

Linear regression discards observations with missing data, so we will remove all such observations from the training and testing sets:

```
pisaTrain = na.omit(pisa2009train)
pisaTest = na.omit(pisa2009test)
```

Because the race variable takes on text values, it was loaded as a factor variable when we read in the dataset with read.csv() – you can see this when you run str(pisaTrain) or str(pisaTest). However, by default R selects the first level alphabetically ("American Indian/Alaska Native") as the reference level of our factor instead of the most common level ("White").

```r
pisaTrain$raceeth = relevel(pisaTrain$raceeth, "White")
pisaTest$raceeth = relevel(pisaTest$raceeth, "White")
```

You can observe this by unning `str` command

```r
str(pisaTrain$raceeth)
```

```
##  Factor w/ 7 levels "White","American Indian/Alaska Native",..: 1 4 5 1 6 5 1 5 1 1 ...
```

Finally, let us build linear model for prediction of readingScore as dependent variable using `lm` function:

```r
lmScore =lm(readingScore ~., data=pisaTrain)
summary(lmScore)
```

```
## 
## Call:
## lm(formula = readingScore ~ ., data = pisaTrain)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -247.44  -48.86    1.86   49.77  217.18 
## 
## Coefficients:
##                                                 Estimate Std. Error
## (Intercept)                                   143.766333  33.841226
## grade                                          29.542707   2.937399
## male                                          -14.521653   3.155926
## raceethAmerican Indian/Alaska Native          -67.277327  16.786935
## raceethAsian                                   -4.110325   9.220071
## raceethBlack                                  -67.012347   5.460883
## raceethHispanic                               -38.975486   5.177743
## raceethMore than one race                     -16.922522   8.496268
## raceethNative Hawaiian/Other Pacific Islander  -5.101601  17.005696
## preschool                                      -4.463670   3.486055
## expectBachelors                                55.267080   4.293893
## motherHS                                        6.058774   6.091423
## motherBachelors                                12.638068   3.861457
## motherWork                                     -2.809101   3.521827
## fatherHS                                        4.018214   5.579269
## fatherBachelors                                16.929755   3.995253
## fatherWork                                      5.842798   4.395978
## selfBornUS                                     -3.806278   7.323718
## motherBornUS                                   -8.798153   6.587621
## fatherBornUS                                    4.306994   6.263875
## englishAtHome                                   8.035685   6.859492
## computerForSchoolwork                          22.500232   5.702562
## read30MinsADay                                 34.871924   3.408447
## minutesPerWeekEnglish                           0.012788   0.010712
## studentsInEnglish                              -0.286631   0.227819
## schoolHasLibrary                               12.215085   9.264884
## publicSchool                                  -16.857475   6.725614
## urban                                          -0.110132   3.962724
## schoolSize                                      0.006540   0.002197
##                                               t value Pr(>|t|)
## (Intercept)                                     4.248 2.24e-05 ***
## grade                                          10.057  < 2e-16 ***
## male                                           -4.601 4.42e-06 ***
## raceethAmerican Indian/Alaska Native           -4.008 6.32e-05 ***
## raceethAsian                                   -0.446  0.65578
## raceethBlack                                  -12.271  < 2e-16 ***
## raceethHispanic                                -7.528 7.29e-14 ***
## raceethMore than one race                      -1.992  0.04651 *
## raceethNative Hawaiian/Other Pacific Islander  -0.300  0.76421
## preschool                                      -1.280  0.20052
## expectBachelors                                12.871  < 2e-16 ***
## motherHS                                        0.995  0.32001
## motherBachelors                                 3.273  0.00108 **
## motherWork                                     -0.798  0.42517
## fatherHS                                        0.720  0.47147
## fatherBachelors                                 4.237 2.35e-05 ***
## fatherWork                                      1.329  0.18393
## selfBornUS                                     -0.520  0.60331
## motherBornUS                                   -1.336  0.18182
## fatherBornUS                                    0.688  0.49178
## englishAtHome                                   1.171  0.24153
## computerForSchoolwork                           3.946 8.19e-05 ***
## read30MinsADay                                 10.231  < 2e-16 ***
## minutesPerWeekEnglish                           1.194  0.23264
## studentsInEnglish                              -1.258  0.20846
## schoolHasLibrary                                1.318  0.18749
## publicSchool                                   -2.506  0.01226 *
## urban                                          -0.028  0.97783
## schoolSize                                      2.977  0.00294 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.81 on 2385 degrees of freedom
## Multiple R-squared:  0.3251, Adjusted R-squared:  0.3172
## F-statistic: 41.04 on 28 and 2385 DF,  p-value: < 2.2e-16
```

Multiple R-squared value of lmScore which is relatively low. This does not necessarily imply that the model is of poor quality. More often than not, it simply means that the prediction problem at hand (predicting a student's test score based on demographic and school-related variables) is more difficult than other prediction problems.

Root mean squared error (RMSE) on the trainin data can be easily calculated with:

```
RMSE = sqrt(mean(lmScore$residuals^2))
RMSE
```

```
## [1] 73.36555
```

Using the `predict` function and supplying the "newdata" argument, we can use the lmScore model to predict the reading scores of students in pisaTest

```
predTest = predict(lmScore, newdata = pisaTest)
summary(predTest)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   353.2   482.0   524.0   516.7   555.7   637.7
```

Next, we can caluclate Sum of squared errors (SSE) and RMSE on test dataset. Note that we have to subtract predicted and real values as follows:

```
SSE = sum((predTest-pisaTest$readingScore)^2)
SSE
```

```
## [1] 5762082
```

```
RMSE = sqrt(SSE/nrow(pisaTest))
RMSE
```

```
## [1] 76.29079
```

As expected, RMSE on test set is somewhat higher. It is interestingly to see the accuracy of our predictions comparing to the baseline model (which always gives the most frequent answer):

```
baseline = mean(pisaTrain$readingScore)
baseline
```

```
## [1] 517.9629
```

Comparing this with men value in predTest, we can see that our model is slightly better. We also can compare sum of squared error on the baseline model (also called total sum of squares - SST):

```
SST = sum((pisaTest$readingScore-mean(pisaTrain$readingScore))^2)
SST
```

```
## [1] 7802354
```

The significant difference between SSE and SST gives us some confident that our model is solid. Finally, we can confirm this by calculatin R-squared in test set:

```
R2=1-SSE/SST
R2
```

```
## [1] 0.2614944
```

This is relatively low value, but as mentioned above, the problem is too complex to be solved with simple technique such as linear regression. Further investigation must include logistic regression, CART models, regression trees or neural networks in finding the best possible model.