Model Building Based on IMR Dataset Using R

Poushali Sengupta, Sumedha Dhar, Debolena Basak, Sunendu Biswas, Aishani Barman Roy, Sumitava Bose, Soumayan Chakraborty, Suchismita Chakrabarty, Arpan Bag (2 Years). Arunodaya Bhattacharya, Pallab Kr. Sinha (5 Years Integrated).

Abstract

Regression helps us not only in understanding the relationship between a dependent variable and another set of variables (namely, independent variables) but also estimating them. However, the analysis of regression stands on some assumptions (such as: normality, homogeneity of variance etc.). So before analyzing the data at hand, we need to check whether it fulfils the assumptions of multiple linear regression.

Now, our main goal is to determine an appropriate subset from the pool of explanatory variables for building the model. Now, while choosing a subset, there are two possible options:

1. Inclusion of irrelevant variables 2. Exclusion of relevant variables

Our search for a quality model lies between these, trying to resolve them.
There are various techniques to judge the performance of a model. We will try to evaluate our model by using some of them, such as: Multiple R-squared, Adjusted R-squared, Akaike's information criterion (AIC), Mallow's Cp.
For the assessment of this method, we think of using algorithms like k-nearest neighborhood and random forest.
k-nearest neighbors algorithm (k-NN) is a non-parametric method used for regression where the input consists of the k closest training examples in the feature space and the output is the property value for the object, which is the average of the values of k nearest neighbors. On the other hand, Random forests conduct regression by constructing a multitude of decision trees at training time and mean prediction (regression) of the individual trees.
After constructing our primary model, we shall try to obtain regression models using these algorithms too, and compare all these three models.

Keywords: Statistics, Data Science, Machine Learning,Linear Regression, Multiple Linear Regression, R squared, forward selection of variables, IMR(infant mortality rate), K- Nearest Neighbors (KNN), Random Forest, PowerPoint, R software, diagnostic plot, Anderson-Darling Test, Breusch Pagan test, Auto correlation, Model Building, response variable, explanatory variable, college project , University of Kalyani.

Introduction

A first step to build a suitable model for our data is to check some essential properties such as if the data is following normality or not, or if there is any autocorrelation or multicollinearity present in the dataset. If there is any autocorrelation or multicollinearity in the dataset, we have to remove some variables from the dataset to make it free from autocorrelation and multicollinearity. Here we apply Anderson-Darling test for checking the normality of the data. To check the autocorrelation, we apply the Durbin-Watson test and for multicollinearity we will check the Variance Inflation Factor(VIF) of each independent variable from the dataset. To make the data multicollinearity free, we have to explore the correlation matrix of the explanatory variables. We allow only those variables which have low variance inflation factor (VIF) in the model.

- **Anderson-Darling Test:** The Anderson–Darling test is a statistical test of whether a given sample of data is drawn from a given probability distribution. In its basic form, the test assumes that there

are no parameters to be estimated in the distribution being tested, in which case the test and its set of critical values is distribution-free. However, the test is most often used in contexts where a family of distributions is being tested, in which case the parameters of that family need to be estimated and account must be taken of this in adjusting either the test-statistic or its critical values. When applied to testing whether a normal distribution adequately describes a set of data, it is one of the most powerful statistical tools for detecting most departures from normality.

- **Durbin-Watson Test:** The Durbin–Watson statistic is a test statistic used to detect the presence of autocorrelation at lag 1 in the residuals (prediction errors) from a regression analysis.

    - Ø **null hypothesis:** the errors on a regression model follow a process with a unit root against
    - Ø **the alternative hypothesis:** that the errors follow a stationary first order auto regression.

- **Variance Inflation Factor (VIF):** The variance inflation factor (VIF) is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. It quantifies the severity of multicollinearity in an ordinary least square regression analysis. It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of collinearity. If the VIF of a variable lies between 1-5, we can't remove the variable from the dataset, but if it lies between 5-10, we may or may not remove it from the dataset. When the VIF of a variable is greater than 10, we must remove it from the dataset.

The second step of this regression diagnostic is to inspect the significance of the regression beta coefficients, as well as, it tells us how well the linear regression model fits to the data. We can use any one of the forward (starting model with no predictors), backward (starting model with all predictors) and stepwise regression process (combination of both forward and backward) to find our suitable model for our dataset.

Here we will apply Stepwise Regression (both forward and backward) to find our suitable model for the dataset where at each step:

- We do not look at every single possible model in the universe that contains k predictors such as in best subset selection but we will just look at the models that contain the k-1 predictors that we already chose in the previous step.
- We just choose the variable that gives the biggest improvement to the model we just had a moment earlier.

After getting the model we evaluate how well the model fits the data. Not only by observing the values of R-squared but also, we take the decision by calculating some other measures as there are many ways to find how well our model fits our dataset. These are:

- **Mallow's cp:** It is used to assess the fit of a regression model that has been estimated using ordinary least squares. It is applied in the context of model selection, where a number of predictor variables are available for predicting some outcome, and the goal is to find the best model involving a subset of these predictors. A small value of C_p means that the model is relatively precise. For a model, if the mallow's cp is very close to the number of Independent variables used in that model, then we can say the model is suitable for the given dataset.
- **AIC (Akaike information criterion):** The Akaike information criterion (AIC) is an estimator of out-of-sample prediction error and thereby relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection. AIC is founded on information theory. When a statistical model is used to represent the process that generated the data, the representation will almost never be exact; so some information will be lost by using the model to represent the process. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit of the model and the simplicity of the model. In other words, AIC deals with both the risk of overfitting and the risk of under fitting. The small value of AIC is preferable.
- **BIC (Bayesian information criterion):** The Bayesian information criterion (BIC) or Schwarz information criterion (also SIC, SBC, SBIC) is a criterion for model selection among a

finite set of models; the model with the lowest BIC is preferred. It is based, in part, on the likelihood function and it is closely related to the Akaike information criterion (AIC).

- **MSE:** The mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. Here small value of MSE is favourable.
- **Partial F statistic**: A partial F-test is an incremental F-test which is used to determine the statistical significance of a group of variables. It is based on two best-fit regression models. It determines the effect of extra variables on the explanatory power by the inclusion of the variables in the equation.
- **PRESS statistic**: The predicted residual error sum of squares (PRESS) statistic is a form of cross-validation used in regression analysis to provide a summary measure of the fit of a model to a sample of observations that were not themselves used to estimate the model. It is calculated as the sums of squares of the prediction residuals for those observations. Lowest value of PRESS statistic indicates best structures.

Now, for example, the linear regression model makes the assumption that the relationship between the predictors (x) and the outcome variable is linear. This might not be true. The relationship could be polynomial or logarithmic.
Additionally, the data might contain some influential observations, such as outliers (or extreme values), that can affect the result of the regression.
Therefore, we should closely diagnose the regression model that we want to build in order to detect potential problems and to check whether the assumptions made by the linear regression model are met or not.
To do so, we generally examine the distribution of **residuals errors** that can tell us more about our data.
Here we shall use Principal Component Analysis for dimension reduction of the dataset.

- **Principal Component Analysis:** Given a collection of points in two, three, or higher dimensional space, a "best fitting" line can be defined as one that minimizes the average squared distance from a point to the line. The next best-fitting line can be similarly chosen from directions perpendicular to the first. Repeating this process yields an orthogonal basis in which different individual dimensions of the data are uncorrelated. These basis vectors are called principal components, and several related procedures principal component analysis (PCA).

To find our primary regression model generated from stepwise regression method, we choose a machine learning algorithm and partition our data into training and testing set where the training set is trained with stepwise regression technique to generate the best subset model based on AIC values. The model is validated on the testing dataset. Here we use the k-fold repeated cross validation technique to validate the model.

k-Fold Cross Validation: Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.
Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.
It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups.
3. For each unique group:
    a. Take the group as a hold out or test data set
    b. Take the remaining groups as a training data set
    c. Fit a model on the training set and evaluate it on the test set

   d. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores.

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

Repeated k-Fold Cross Validation is the technique to repeat the k-Fold Cross Validation for more than one time.

In our work, we build other two models using K-Nearest Neighborhood and Random Forest methods and compare them with our primary model generated by Stepwise Regression method.

- **K-Nearest Neighbors:** Nearest neighbor search (NNS), as a form of proximity search, is the optimization problem of finding the point in a given set that is closest (or most similar) to a given point. Closeness is typically expressed in terms of a dissimilarity function: the less similar the objects, the larger the function values. K-NN identifies the top *k* nearest neighbors to the query. This technique is commonly used in predictive analytics to estimate or classify a point based on the consensus of its neighbors. *k*-nearest neighbor graphs are graphs in which every point is connected to its *k* nearest neighbors.

- **Random Forest:** Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Here,

- We begin by replacing the outliers with other variables in such a way, that the dataset becomes free of outliers.

- Then we build a full model on the dataset where the IMR values depend on remaining all predictors of the dataset.
- Then we check the autocorrelation.
- Next we check the multicollinearity.
- Then partition the dataset in 80:20, where training set contains 80% of the data and testing set contains 20% of the data.
- After that train the dataset with stepwise regression and evaluate the model on testing dataset.
- Then validate the model with 10-fold repeated cross validation for 100 times and find out the final model.
- Now apply principal component analysis to check if there is any need of dimension reduction.
- Calculate the mallow's cp of the final model to check how good the model fits to the data.
- Then we explain **residuals errors** and **fitted values**.
- Next, we present linear **regression assumptions**, as well as, potential problems we can face when performing regression analysis.
- After that we check the normality of the residuals of the final model by using Anderson-Darling test.
- Then we do the Breusch Pagan test to check if our model follow homoscedasticity or not.
- Finally, we describe some built-in **diagnostic plots** in R for testing the assumptions underlying linear regression model.
- To compare our final model, build other two models based on k-nearest neighbor and Random Forest.
- At last compare those all three models and take the decision if our first model is suitable for our dataset or not.

Data Description:

We use the data set IMR (Infant Mortality Rate) dataset which we have collected from the website data.gov.in.

Infant mortality rate (IMR) is defined in the following way:

IMR=1000 * (D_0/B)

Where, D_0= number of deaths among children of age 0 l.b.d

B= number of live births

Now, IMR of a state may be controlled by variables related to the health sector. For example, if pregnant women go through check-ups, i.e., they are observed regularly by doctors, intuitively, infant mortality should decrease. Or, if in a state, if there is a trend of home deliveries, instead of institutional deliveries (i.e. in a hospital) that should have a positive effect on infant mortality.

But, there are many more factors that should influence IMR. For example, in a state having a higher literacy rate, people should be more aware than a state with a lower one. Or, in a state with better transport systems, people can reach hospitals or health centres more easily, but if in a state, the transport system is not that good, that should negatively influence mortality.

So, we decide to take not only the data directly related to health facilities, we consider many other variables, which are related to a state's development, and want to see if they have a significant effect on IMR.

We write the variables names and explain what they denote here-

1. **ANC:** Percentage of ANC-registered women (Antenatal care) who received 3 ANC check-ups
2. **Newborn:** percentage of new-born children visited by doctors within 24 hours of home delivery
3. **Popden:** population density of a state
4. **Sexratio:** no. of females per thousand males
5. **Maleliterate:** male literacy rate
6. **Femalelierate**: female literacy rate
7. **Highway:** highway length per square kilometre
8. **Electricitycon:** per capita consumption of electricity in kWh
9. **Rail route:** total rail route per sq. km
10. **Ruralteledensity:** tele density is the number of telephone connections for every hundred individuals living within an area. Rural teledensity denotes this for rural areas in each state
11. **Urbanteledensity:** same for urban areas in each state

12. **Unemployrate:** unemployment rate of a state, which is defined by (unemployed people/labour force) *100. Labour force is the sum of unemployed and employed persons.

13. **GERprimary:** total enrolment in primary education (grades I-IV), regardless of age, expressed as a percentage of the eligible official primary school age population.

14. **GERupprimary:** total enrolment in upper primary education (grades V-VIII), regardless of age, expressed as a percentage of the eligible official upper primary school age population.

15. **Govthospital:** no. of people served by each govt. hospital

16. **SC:** no. of sub centres in each state

17. **PHC:** no. of primary health centres in each state

18. **CHC:** no. of community health centres in each state

19. **Ruralexpend:** average expenditure for a child for treatment during his stay at hospitals in rural areas

20. **Urbanexpend:** average expenditure for a child for treatment during his stay at hospitals in urban areas

Analysis of Data:

In this section we will discuss the step by step analysis that we have done by using R. All the outputs and results are described here elaborately.
We are going to work on the IMR dataset mentioned above.

- **[Removing Outliers]** The most important thing is to remove outliers from the dataset. Here instead of removing the outliers from the datasets, we replace them with other values in such a way, that the dataset becomes free of outliers. To replace the outliers from the variables that is greater than (Q3 + 1.5*IQR) with maximum value of the variable without outliers and less than (Q1 - 1.5*IQR) with minimum value of the variable without outliers. 1st we remove the categorical variable 'state' from the data then we create boxplot of the data to check if the variables contains any outliers. Then we study the box plot given for all 21 variables and find out that all the outliers for the variables is greater than (Q3 + 1.5*IQR) so, we replace the outliers for the variables containing outliers with the maximum value of the variables without outliers.
- **[Building Full Model]** Then we build a full model where the IMR values depend on remaining variables of the dataset.

- **[Checking Normality]** After that, we run Anderson-darling normality test on the dataset to check whether it fulfils normality assumption. The P-value being 0.8599, we can accept the null hypothesis of the dataset having normality of the residuals of the full model.

- **[Checking Autocorrelation]** Then we conduct the Durbin-Watson test to check whether there is any autocorrelation. The P-value being 0.332, we have to say that there is no positive autocorrelation present in the dataset.

- **[Removing Multicollinearity]** Now, we are going to study the multicollinearity between the explanatory variables. At first, we are going to check the pairwise correlations of the variables, who come from the same class. Here we use the built in function ggcorrplot() in R to plot the

correlations of the variables of the dataset. It helps us to detect the high correlations among the variables and remove the multicollinearity from the dataset.

For example,

1. sub-centres (SC), primary health centre(PHC), and community health centre(PHC) are of the same kind. Their correlations with IMR are 0.43, 0.4, 0.43 respectively. On the other hand, the correlation coefficient between PHC and CHC is 0.87 and that of between CHC and SC is 0.89. So, we can see that multicollinearity exists here, and to remove it we keep CHC, we can throw out SC and PHC.

2. GERupprimary and GERprimary are highly correlated (0.79). But, the first variable has a correlation coefficient of -0.45 with IMR, compared to -0.15 for the later one. So, we can drop GERprimary.

3. Maleliterate and femalelierate are highly correlated, so we can drop maleliterate.

4. Ruralexpend and urbanexpend are highly correlated with each other. So we can drop urbanexpend as it has less correlation with IMR than ruralexpend.

5. Keeping ANC, we can drop newborn going by the same logic.

In this way, we finally get a subset of variables which are: ANC, sexratio, femaleliterate, highway, electricconsumption, rural teledensity, urban teledensity, GERupprimary, CHC, ruralexpend, govthospital.

We calculate VIF of each of them and see that each of the variables has a VIF less than 5. So we can safely say that this set of remaining variables does not suffer from severe multicollinearity.

- **[Principal Component Analysis]** Now we are going to create a full model by regressing this final set of variables on IMR. We want to see whether more reduction of dimension is possible, using principal component analysis. We see that every variable is carrying significant information here, so we cannot drop any more variables. Therefore, we accept this set of variables as our final set.

- **[Stepwise Regression and Model Validation]** Next, we want to build a stepwise regression model on this set of variables using a machine learning algorithm. For this, we are dividing our dataset into two parts (using random sampling), namely training dataset and testing dataset such that the ratio of training and testing dataset is 80:20. We apply a stepwise regression on the training dataset, where the subset model with least AIC is selected and validated on the testing dataset. This whole procedure (i.e., from sampling to model selection) is repeated for 1000 times with 10-fold repeated cross-validation. At last we get the final model and validate the model on the testing dataset. By doing this, we obtain multiple R-squared of 0.84.
  The final model is:

  IMR =25.0517-1.25*femalelitrate-0.85*GERupprimary+0.026*govthospital-0.14*rurexpnd.

  Mallow's Cp of this model is:4.325234, which is close enough to the number of predictors. So we can say that the model is good. We study the diagnostic plots and various tests (Anderson-Darling, Breusch Pagan) and see that the model follows all the assumptions of linear regression (e.g., linearity, homogeneity of variance, normality of residuals).

  This is our first model. We shall construct two more models, using k-nearest neighbor algorithm and random forest and compare with this model.

- **[k-Nearest Neighbor Model]** Here for constructing the K-NN model we divide the dataset into two parts, like the previous case. We train the dataset using k-nn method, where we have

used the function expand.grid() as the tune grid, varying the values of k (where k is the tuning parameter) from 1 to 3 (because we have taken square root of the number of variables, and square root of 11 is 3 approximately. This whole method is repeated 1000 times, with 10-fold cross-validation. Here we obtain multiple R-squared 0.846.

The final model obtained is the 3-nearest neighbor model. The variables in descending order with respect to variable importance to the R-squared are: femaleliterate (100), highway (77), rural teledensity (76), rurexpend (68), CHC (48), electricitycon (37), urban teledensity (22), ANC (7), GERupprimary (5), sexratio (3), govthospital (0).

- **[Random Forest Model]** For constructing the random forest model, we divide the dataset as usual like the previous two cases. We train the training dataset using random forest method where we again use the function expand.grid(). We vary the tuning parameter mtry from 1 to 15 and take the number of trees 1000. This procedure is repeated 1000 times with 10-fold cross-validation with searching parameter 'random'. When mtry value is 10, we obtain the final model with multiple R-squared 0.87.
  The variables in descending order with respect to variable importance to the R-squared are: femalelitrate (100), rurexpend (43), sex ratio (21), CHC (15), rural teledensity (9), electricitycon (5), GERupprimary (3), highway (2),govthospital (1), ANC (0.8), urban tele density (0).

- **[Final Decision]** The model we obtained from stepwise regression is a decent model to predict IMR, as we can see that the multiple R-squared is high. However, we can see that multiple R-squared of three models are almost same. So, we can surely say the model we constructed at first is suitable and robust for our dataset. But one can use any of these three models according to his wish.

Conclusion and Future Scope:

Several techniques are available to select a subset model and our main focus was on stepwise regression technique.
Without model validation, we cannot know whether our model works well. Now, it's true that stepwise regression does not guarantee the best subset model. So, we took help of machine learning algorithms like nearest neighbor and random forest, for a comparative study. We can safely conclude that the performance of our primary model, i.e.,
IMR= 25.0517-1.25*femalelitrate-0.85*GERupprimary+0.026*govthospital-0.14*rurexpnd, is pretty good.
We wish to make use of this dataset for forecasting in future, using proper machine learning algorithms.

Appendix

Loading Required R packages

```
> library(caret)
```
Tools for Building Machine Learning Algorithm.

```
> library(olsrr)
```
Tools for Building OLS Regression Models

```
>library(MASS)
```
Functions and datasets to support Venables and Ripley, "Modern Applied Statistics with S"
```
>library(broom)
```
creates a tidy data frame from statistical test results.
```
>library(nortest)
```
To check autocorrelation in the dataset.

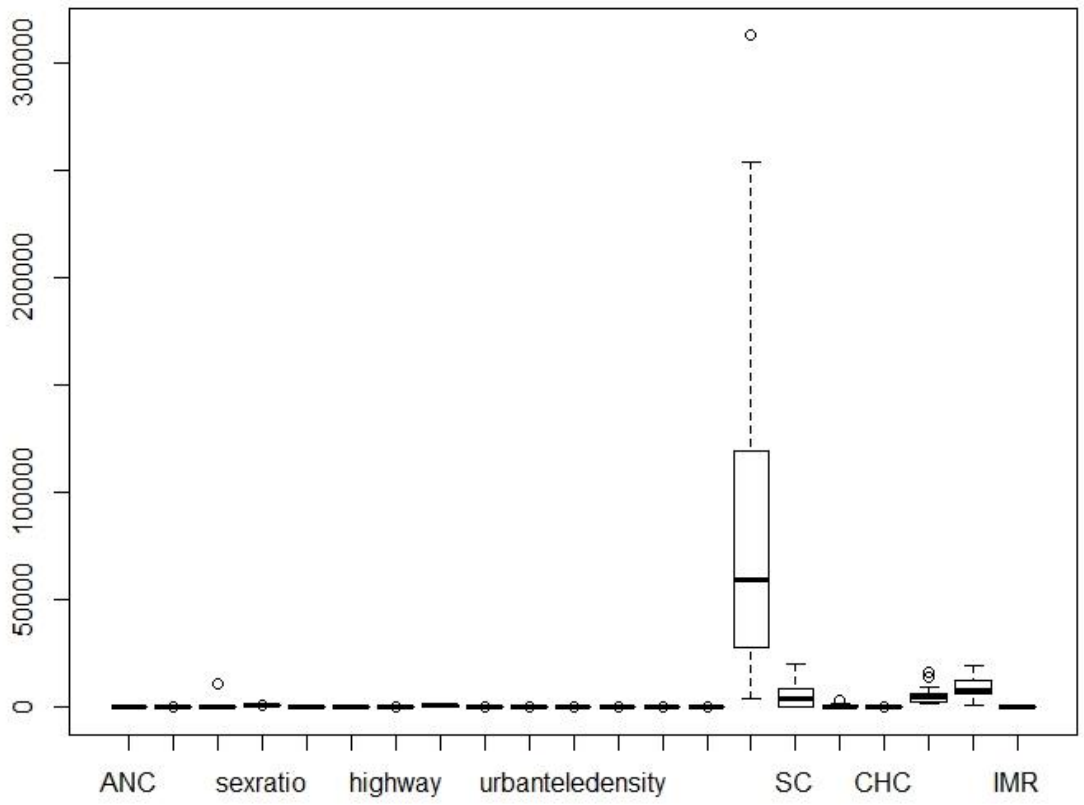
```
>library(car)
```
For Principal Component Analysis.

Loading Dataset:

```
> setwd("C:/Users/Poushali Sengupta/Desktop/research project/GD/")
> data<-read.csv("data.csv",header = T)
> data
> head(data)
b       3931  22
> summary(data)
       state         ANC           Newborn          popden          sexratio       malelitrate
 an.pradesh: 1   Min.   :32.57   Min.   : 23.57   Min.   :   17.0   Min.   : 866.0   Min.   :73.40
 ar.pradesh: 1   1st Qu.:52.74   1st Qu.: 44.26   1st Qu.:  132.0   1st Qu.: 919.0   1st Qu.:79.20
 assam     : 1   Median :73.89   Median : 62.70   Median :  308.0   Median : 950.0   Median :82.90
 bihar     : 1   Mean   :66.91   Mean   : 62.01   Mean   :  739.7   Mean   : 948.7   Mean   :84.06
 chattisgar: 1   3rd Qu.:78.72   3rd Qu.: 74.41   3rd Qu.:  551.0   3rd Qu.: 976.0   3rd Qu.:88.30
 delhi     : 1   Max.   :85.81   Max.   :123.47   Max.   :11297.0   Max.   :1084.0   Max.   :96.00
 (Other)   :23
femalelitrate      highway        electricitycon     rail.route      ruralteledensity urbanteledensity
Min.   :52.70   Min.   :0.00600   Min.   : 145.4   Min.   :0.00000   Min.   : 29.4    Min.   :116.3
1st Qu.:60.00   1st Qu.:0.02100   1st Qu.: 593.9   1st Qu.:0.00100   1st Qu.: 39.9    1st Qu.:131.5
Median :70.70   Median :0.03000   Median : 981.9   Median :0.01700   Median : 42.7    Median :153.0
Mean   :69.77   Mean   :0.03228   Mean   : 979.0   Mean   :0.05064   Mean   : 53.6    Mean   :154.5
3rd Qu.:76.40   3rd Qu.:0.03800   3rd Qu.:1297.3   3rd Qu.:0.03700   3rd Qu.: 57.4    3rd Qu.:154.3
Max.   :92.00   Max.   :0.07300   Max.   :2045.0   Max.   :0.80500   Max.   :226.9    Max.   :325.9

 unemployrate      GERprimary     GERupprimary     govthospital          SC             PHC
Min.   : 0.700   Min.   : 84.9   Min.   : 72.40   Min.   :  4321   Min.   :   34   Min.   :   6.0
1st Qu.: 1.500   1st Qu.:100.5   1st Qu.: 90.90   1st Qu.: 27874   1st Qu.:  420   1st Qu.: 105.0
Median : 2.500   Median :104.0   Median : 98.30   Median : 59682   Median : 4209   Median : 485.0
Mean   : 4.707   Mean   :108.4   Mean   : 99.02   Mean   : 85596   Mean   : 5073   Mean   : 822.9
3rd Qu.: 3.600   3rd Qu.:113.3   3rd Qu.:105.80   3rd Qu.:119033   3rd Qu.: 8703   3rd Qu.:1293.0
Max.   :37.000   Max.   :149.2   Max.   :138.80   Max.   :312778   Max.   :20521   Max.   :3578.0

     CHC          rurexpnd       urbanexpnd         IMR
Min.   :  0   Min.   : 1529   Min.   :  958   Min.   : 9.00
1st Qu.: 25   1st Qu.: 2631   1st Qu.: 6028   1st Qu.:24.00
Median :117   Median : 4742   Median : 8075   Median :35.00
Mean   :148   Mean   : 5484   Mean   : 9387   Mean   :33.41
3rd Qu.:221   3rd Qu.: 6488   3rd Qu.:12488   3rd Qu.:42.00
Max.   :579   Max.   :16351   Max.   :19477   Max.   :54.00
```

Checking Outliers:

```
> boxplot(data1)
```

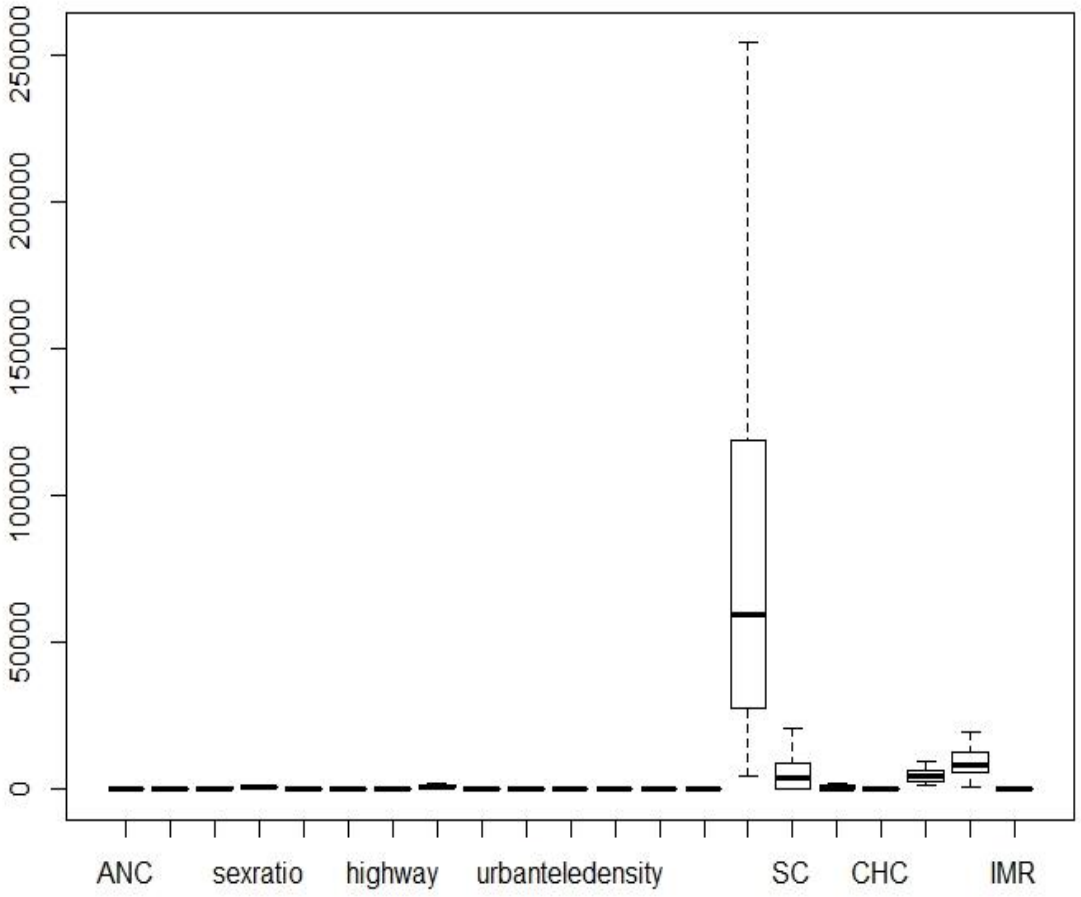


We can see there are some outliers and we need to remove it.

Removing Outliers:

```
> for(i in 1:21){
+   outliers<-boxplot(data1[i], plot = FALSE)$out
+   if(length(outliers)>0){
+     x<-data1
+     x<- x[-which(data1[,i] %in% outliers),]
```

```
+      n=length(data1$Newborn)
+      for(j in 1:n){
+        if(data1[i][j,]>=outliers[1]){
+          data1[i][j,]=max(x[i])
+        }
+      }
+
+    }
+ }
> boxplot(data1)
```



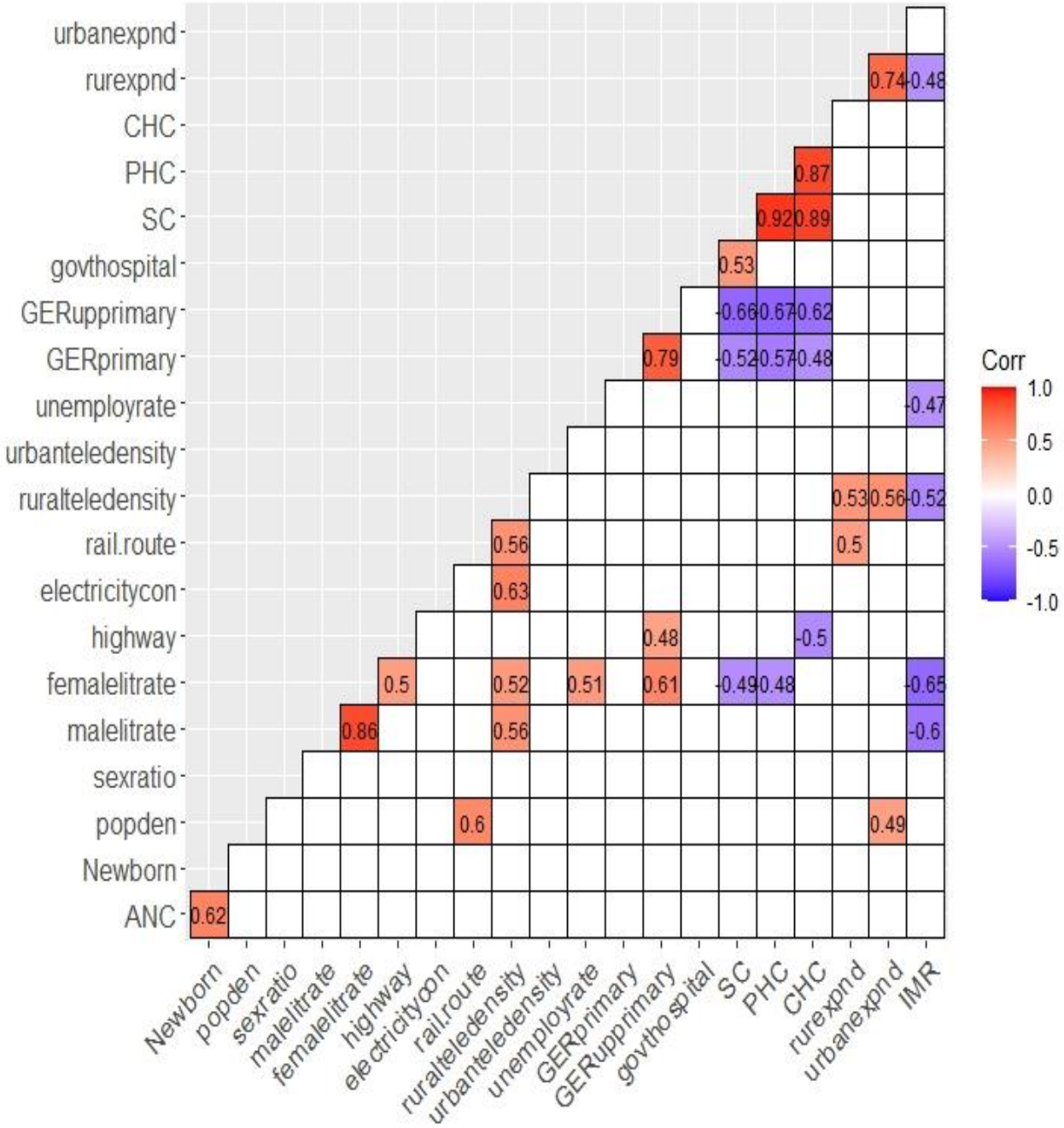
Checking Autocorrelation:

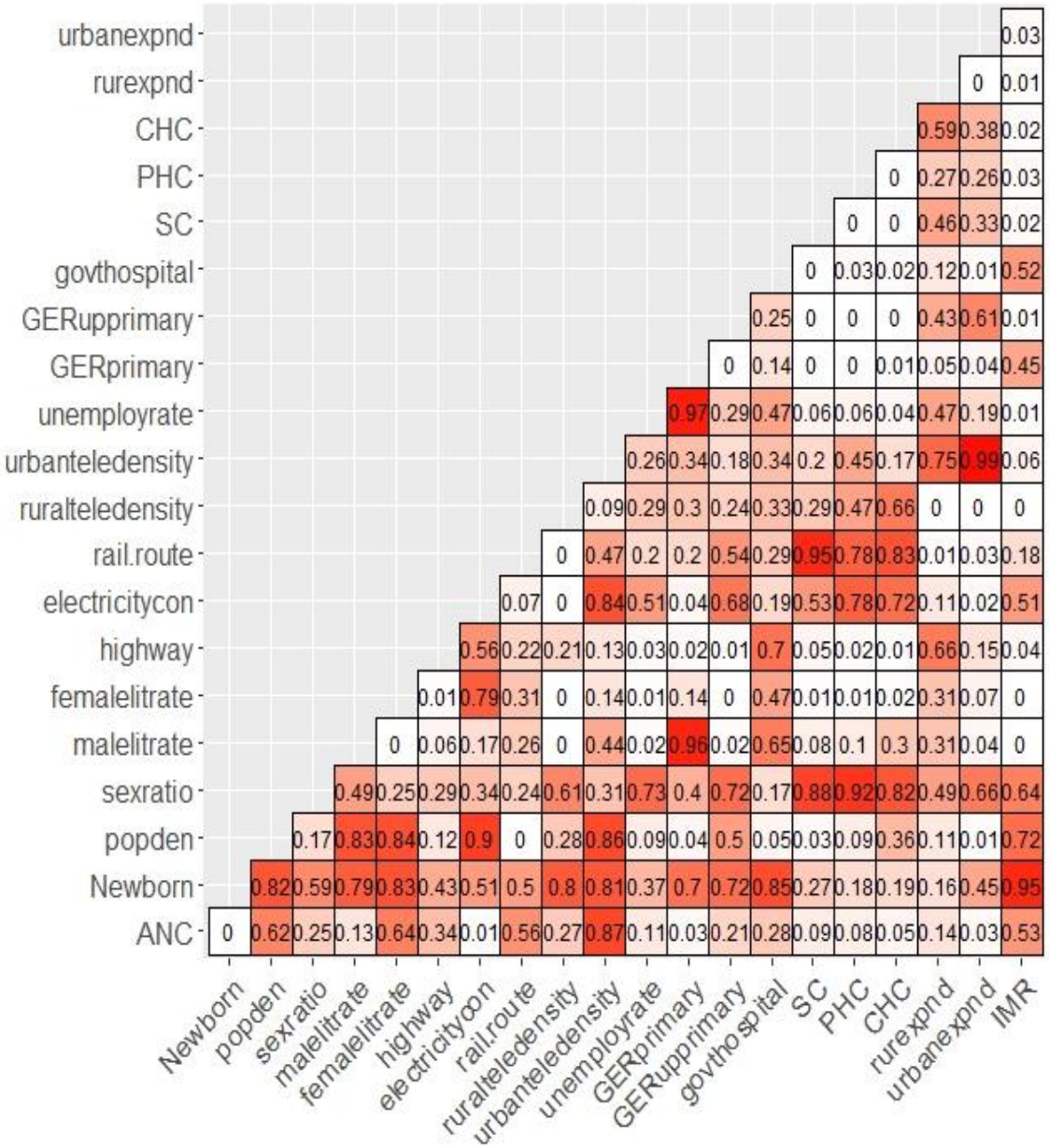
```
> dwtest(fullmodel)

        Durbin-Watson test

data:  fullmodel
DW = 1.8537, p-value = 0.332
alternative hypothesis: true autocorrelation is greater than 0
```

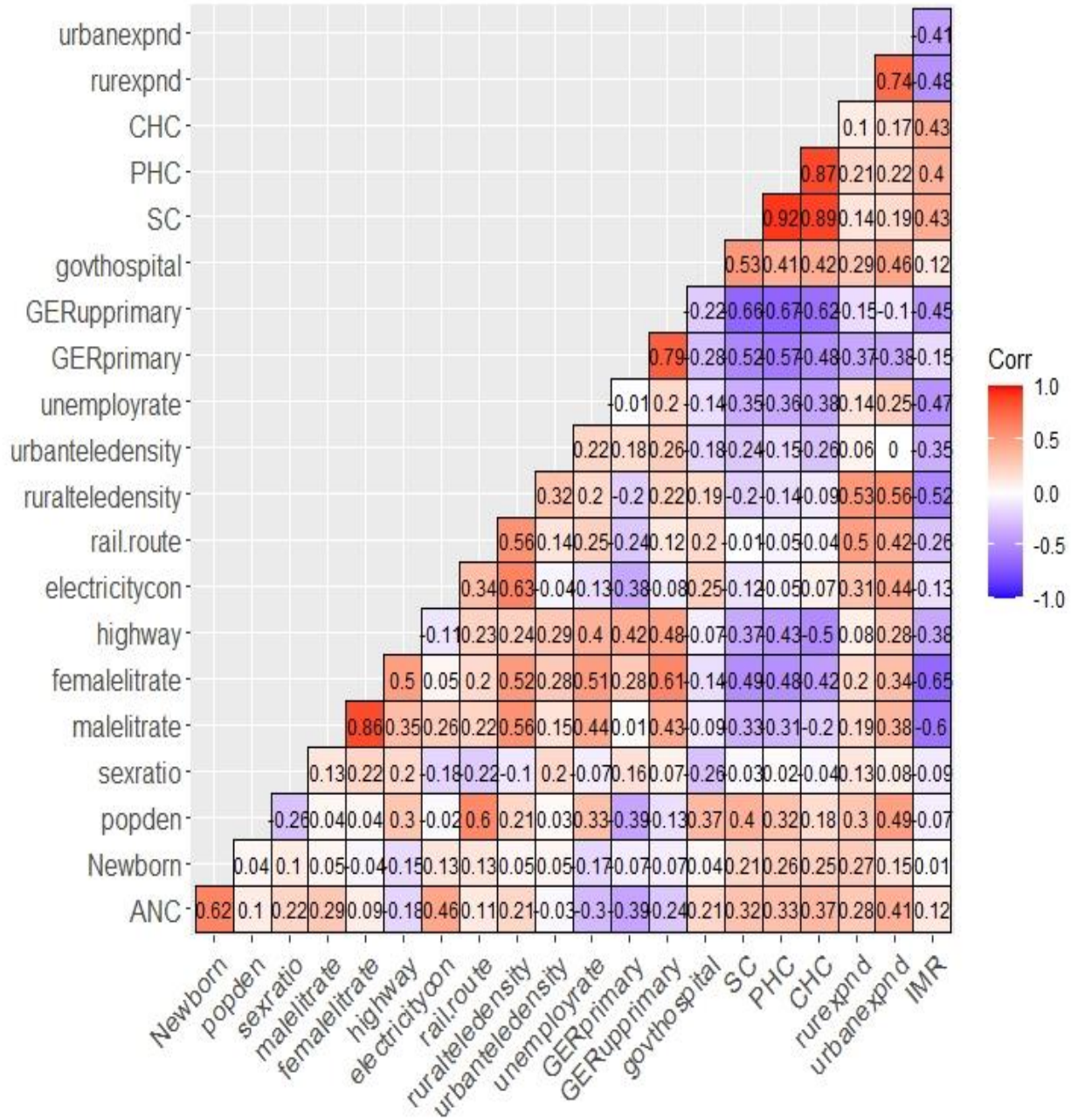
Checking Multicolinearity:

```
> vif(fullmodel)
            ANC         Newborn           popden        sexratio     malelitrate
       7.249528        3.108516       12.208119        2.712198       27.289735
  femalelitrate         highway  electricitycon      rail.route ruralteledensity
      16.183662       12.363451        7.865841        4.678501        7.991672
urbanteledensity     unemployrate      GERprimary    GERupprimary     govthospital
       1.969492        4.418525       56.290339       38.064116        3.099289
            SC             PHC             CHC         rurexpnd       urbanexpnd
      29.597638       15.854466       33.052644        5.704737        8.623010
```

```
> ggcorrplot(y,hc.order = F,type ="lower",lab=T,lab_size =3.2,
+            lab_col = "black",outline.color = "black",
+            p.mat=cor_pmat(data1), sig.level = 0.01,insig="blank" ,
+            show.legend = T,ggtheme = theme_grey())
> t=cor_pmat(data1)
> #par(mfrow=c(1,2))
> ggcorrplot(t,hc.order = F,type ="lower",lab=T,lab_size =3.2,
+            lab_col = "black",outline.color = "black",
+            show.legend = F,ggtheme = theme_grey())
> ggcorrplot(y,hc.order = F,type ="lower",lab=T,lab_size =3.2,
+            lab_col = "black",outline.color = "black",
+            show.legend = T,ggtheme = theme_grey())
```

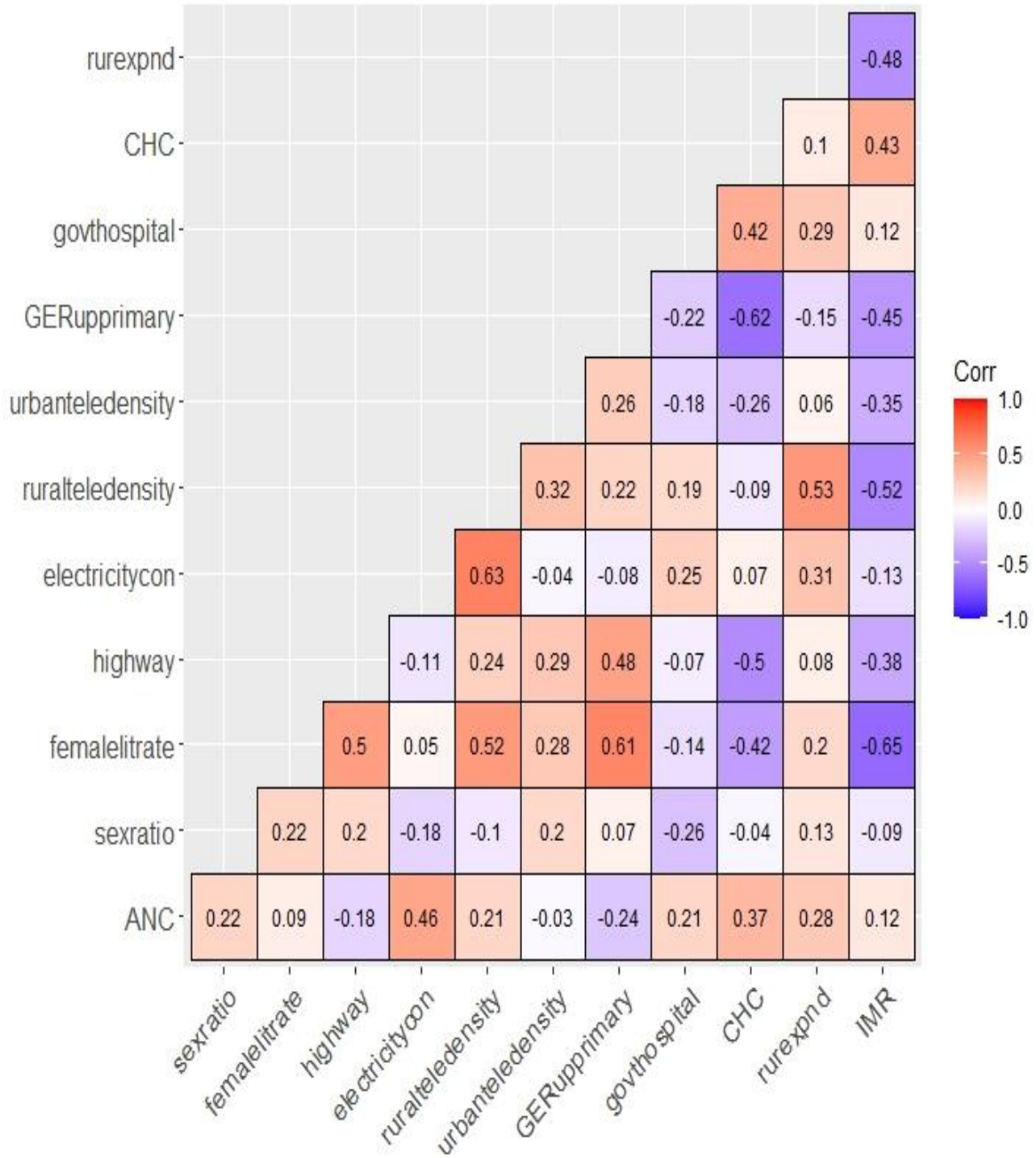

urbanexpnd
rurexpnd
CHC
PHC
SC
govthospital
GERupprimary
GERprimary
unemployrate
urbanteledensity
ruralteledensity
rail.route
electricitycon
highway
femalelitrate
malelitrate
sexratio
popden
Newborn
ANC
0.74 -0.48
0.87
0.92 0.89
0.53
-0.66 0.67 0.62
0.79 -0.52 0.57 0.48
-0.47
0.53 0.56 -0.52
0.56
0.5
0.63
0.48
-0.5
0.5
0.52
0.51
0.61
-0.49 0.48
-0.65
0.86
0.56
-0.6
0.6
0.49
0.62
Newborn
popden
sexratio
malelitrate
femalelitrate
highway
electricitycon
rail.route
ruralteledensity
urbanteledensity
unemployrate
GERprimary
GERupprimary
govthospital
SC
PHC
CHC
rurexpnd
urbanexpnd
IMR
Corr
1.0
0.5
0.0
-0.5
-1.0

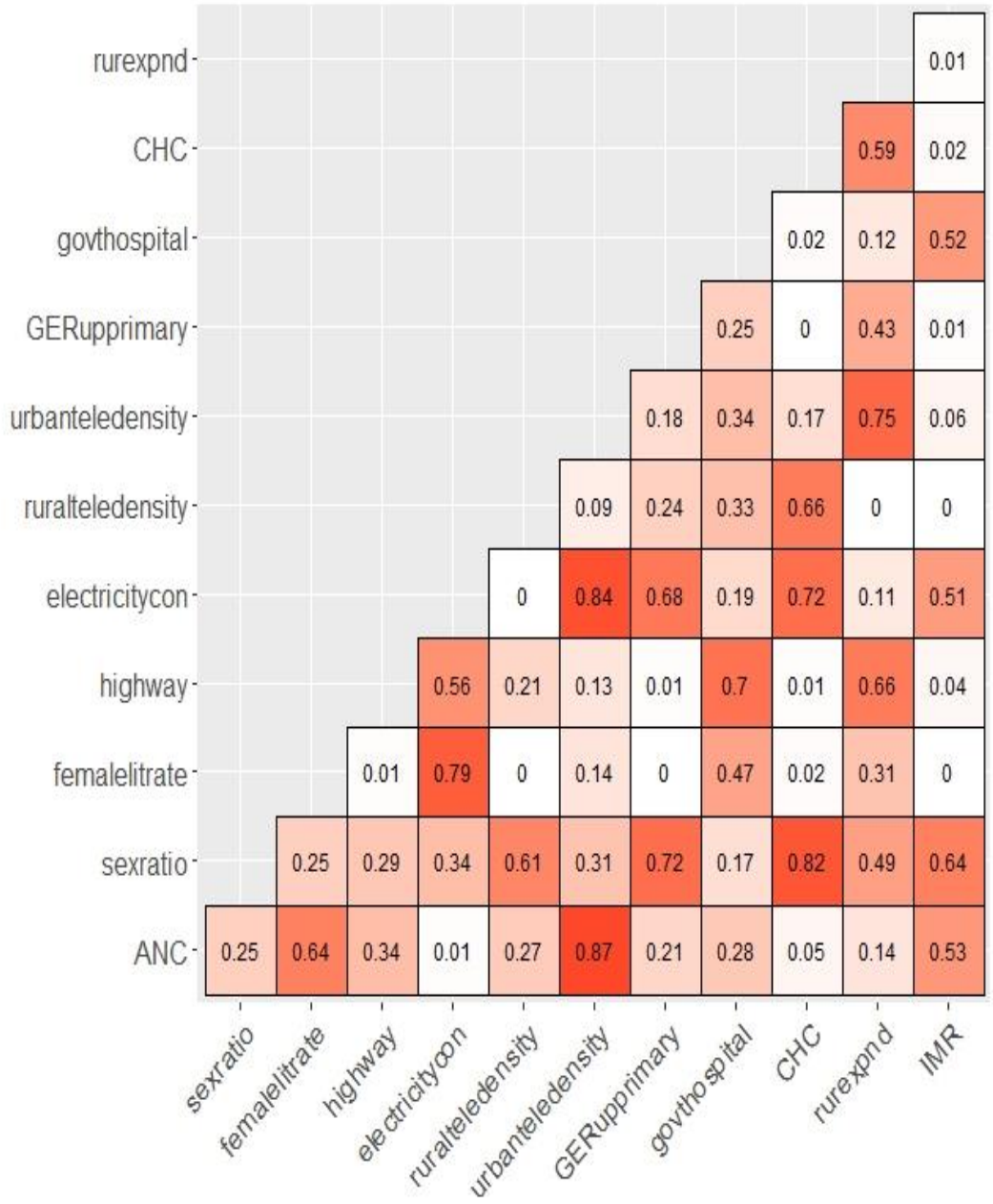

urbanexpnd 0.03
rurexpnd 0 0.01
CHC 0.59 0.38 0.02
PHC 0 0.27 0.26 0.03
SC 0 0 0.46 0.33 0.02
govthospital 0 0.03 0.02 0.12 0.01 0.52
GERupprimary 0.25 0 0 0 0.43 0.61 0.01
GERprimary 0 0.14 0 0 0.01 0.05 0.04 0.45
unemployrate 0.97 0.29 0.47 0.06 0.06 0.04 0.47 0.19 0.01
urbanteledensity 0.26 0.34 0.18 0.34 0.2 0.45 0.17 0.75 0.99 0.06
ruralteledensity 0.09 0.29 0.3 0.24 0.33 0.29 0.47 0.66 0 0 0
rail.route 0 0.47 0.2 0.2 0.54 0.29 0.95 0.78 0.83 0.01 0.03 0.18
electricitycon 0.07 0 0.84 0.51 0.04 0.68 0.19 0.53 0.78 0.72 0.11 0.02 0.51
highway 0.56 0.22 0.21 0.13 0.03 0.02 0.01 0.7 0.05 0.02 0.01 0.66 0.15 0.04
femalelitrate 0.01 0.79 0.31 0 0.14 0.01 0.14 0 0.47 0.01 0.01 0.02 0.31 0.07 0
malelitrate 0 0.06 0.17 0.26 0 0.44 0.02 0.96 0.02 0.65 0.08 0.1 0.3 0.31 0.04 0
sexratio 0.49 0.25 0.29 0.34 0.24 0.61 0.31 0.73 0.4 0.72 0.17 0.88 0.92 0.82 0.49 0.66 0.64
popden 0.17 0.83 0.84 0.12 0.9 0 0.28 0.86 0.09 0.04 0.5 0.05 0.03 0.09 0.36 0.11 0.01 0.72
Newborn 0.82 0.59 0.79 0.83 0.43 0.51 0.5 0.8 0.81 0.37 0.7 0.72 0.85 0.27 0.18 0.19 0.16 0.45 0.95
ANC 0 0.62 0.25 0.13 0.64 0.34 0.01 0.56 0.27 0.87 0.11 0.03 0.21 0.28 0.09 0.08 0.05 0.14 0.03 0.53
Newborn
popden
sexratio
malelitrate
femalelitrate
highway
electricitycon
rail.route
ruralteledensity
urbanteledensity
unemployrate
GERprimary
GERupprimary
govthospital
SC
PHC
CHC
rurexpnd
urbanexpnd
IMR



After removing the variables with high VIF, we get the following result:

```
> data2<-data1[,-c(2,3,5,9,12,13,16,17,20)]
> data2
> y1=cor(data2)
> ggcorrplot(y1,hc.order = F,type ="lower",lab=T,lab_size =3.2,
+            lab_col = "black",outline.color = "black",
+            show.legend = T,ggtheme = theme_grey())
> t1=cor_pmat(data2)
> ggcorrplot(t1,hc.order = F,type ="lower",lab=T,lab_size =3.2,
+            lab_col = "black",outline.color = "black",
+            show.legend = F,ggtheme = theme_grey())
> fullmodel2<-lm(IMR~., data = data2)
```


rurexpnd
CHC
govthospital
GERupprimary
urbanteledensity
ruralteledensity
electricitycon
highway
femalelitrate
sexratio
ANC
-0.48
0.1 0.43
0.42 0.29 0.12
-0.22 -0.62 -0.15 -0.45
0.26 -0.18 -0.26 0.06 -0.35
0.32 0.22 0.19 -0.09 0.53 -0.52
0.63 -0.04 -0.08 0.25 0.07 0.31 -0.13
-0.11 0.24 0.29 0.48 -0.07 -0.5 0.08 -0.38
0.5 0.05 0.52 0.28 0.61 -0.14 -0.42 0.2 -0.65
0.22 0.2 -0.18 -0.1 0.2 0.07 -0.26 -0.04 0.13 -0.09
0.22 0.09 -0.18 0.46 0.21 -0.03 -0.24 0.21 0.37 0.28 0.12
sexratio
femalelitrate
highway
electricitycon
ruralteledensity
urbanteledensity
GERupprimary
govthospital
CHC
rurexpnd
IMR
Corr
1.0
0.5
0.0
-0.5
-1.0




```
> fullmodel2<-lm(IMR~., data = data2)
> vif(fullmodel2)
             ANC          sexratio    femalelitrate          highway    electricitycon
        2.095973          1.577815          3.282077          1.794383          3.510487
ruralteledensity urbanteledensity     GERupprimary     govthospital              CHC
        5.558033          1.506385          2.467746          1.606688          2.711272
        rurexpnd
        1.968191
```

Principal Component Analysis

```
> data2.pca<-prcomp(data2,center = TRUE, scale. = TRUE)
> data2.pca
Standard deviations (1, .., p=12):
 [1] 1.9059903 1.6143994 1.1590311 0.9748307 0.9219216 0.8778366 0.7587782 0.6234888 0.601817
[10] 0.4835562 0.4654886 0.2637373

Rotation (n x k) = (12 x 12):
                         PC1         PC2          PC3         PC4         PC5         PC6
ANC              -0.07423762  0.39540840 -0.410701823  0.37412422 -0.27798423  0.10380582
sexratio          0.11568540 -0.05782512 -0.763122714 -0.07000634 -0.14259956 -0.04835337
femalelitrate     0.43450538  0.07095567 -0.078095295  0.05468932 -0.30496670 -0.04748827
highway           0.36046575 -0.10098545 -0.009729076 -0.25087336 -0.32723435  0.32703426
electricitycon    0.04116652  0.46470203  0.187899060  0.49812735  0.01266003 -0.06126811
ruralteledensity  0.29630278  0.42903136  0.163151003  0.11536637  0.14047948  0.07935024
urbanteledensity  0.27328499 -0.01461020 -0.196593225  0.05671448  0.63580996  0.63486886
GERupprimary      0.38858249 -0.18310401  0.163824099  0.18216702 -0.31218335  0.08490135
govthospital     -0.13893176  0.34588778  0.256398363 -0.43074642 -0.34140358  0.45072378
CHC              -0.36273860  0.26049968 -0.171128823 -0.16987033  0.00166116  0.17033777
rurexpnd          0.13758477  0.43326724 -0.133489381 -0.47617788  0.15897617 -0.33721434
IMR              -0.42118076 -0.12688709 -0.050180257  0.22409021 -0.19713708  0.33388984
                         PC7         PC8          PC9         PC10         PC11         PC12
ANC              -0.083545176  0.01728692  0.5882267041  0.09979537 -0.106968398 -0.25836898
sexratio          0.152823173 -0.30555233 -0.4713897034 -0.14709799  0.055059826 -0.09873447
femalelitrate    -0.428564690  0.31264175  0.0215369837 -0.30977000  0.371629602  0.43113611
highway           0.552342280  0.42847272  0.0643131372  0.05837014 -0.293694195  0.01817012
electricitycon    0.366439697 -0.16449185 -0.2472489377 -0.20758402 -0.244037414  0.41218119
ruralteledensity -0.004232072  0.25261423 -0.3836400004  0.19495746  0.325043584 -0.55453125
urbanteledensity -0.084566192 -0.11991511  0.1364157135 -0.01333361  0.001935465  0.18868441
GERupprimary     -0.293071238 -0.40705297 -0.1361990303  0.56663963 -0.237531355  0.05614330
govthospital     -0.079961347 -0.43182498 -0.0003936958 -0.28205249  0.111275706 -0.07256724
CHC              -0.412300241  0.38266083 -0.3646899736  0.18170170 -0.445987609  0.17905830
rurexpnd          0.158566180 -0.11817123  0.1910705745  0.45163830  0.155976523  0.32538707
IMR               0.228142281  0.08482547 -0.1104674824  0.38747191  0.551283002  0.27916257
> summary(data2.pca)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9
Standard deviation     1.9060 1.6144 1.1590 0.97483 0.92192 0.87784 0.75878 0.62349 0.60182
Proportion of Variance 0.3027 0.2172 0.1119 0.07919 0.07083 0.06422 0.04798 0.03239 0.03018
Cumulative Proportion  0.3027 0.5199 0.6319 0.71106 0.78189 0.84611 0.89408 0.92648 0.95666
                          PC10    PC11   PC12
Standard deviation     0.48356 0.46549 0.2637
Proportion of Variance 0.01949 0.01806 0.0058
Cumulative Proportion  0.97615 0.99420 1.0000
>
```

Stepwise Regression and Model Validation

```
> set.seed(1234)
> ran<-sample(1:nrow(data2),0.8*nrow(data2))
> ran
 [1]  4 18 17 27 22 16  1  6 14 11 21 10  5 15 26 12 29 13  3 20 28  9  2
> training <- data2[ran,]
> test <- data2[-ran,]
> View(training)
> training <- data2[ran,]
> test <- data2[-ran,]
> View(training)
> #trcontrol<-trainControl(method = 'oob')
> trcontrol<-trainControl(method = 'repeatedcv', number = 10,
 repeats =1000,search = "random")
> set.seed(185)
> fit <- train(IMR ~ ., data = training, tuneLength= 15 ,
method = 'lmStepAIC',metric= "Accuraccy", trControl=trcontrol)
> fit
Linear Regression with Stepwise Selection

23 samples
11 predictors

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 1000 times)
Summary of sample sizes: 21, 21, 21, 20, 21, 20, ...
Resampling results:

  RMSE      Rsquared   MAE
  14.88681  0.8384254  12.98719

> varImp(fit)
loess r-squared variable importance

                 Overall
femalelitrate    100.000
rurexpnd          89.646
ruralteledensity  77.982
CHC               52.803
urbanteledensity  38.520
sexratio          38.009
GERupprimary      35.619
govthospital      32.176
highway           28.476
ANC                7.476
electricitycon     0.000
>
```

```
> varImp(fit)
loess r-squared variable importance

                 Overall
femalelitrate    100.000
rurexpnd          89.646
ruralteledensity  77.982
CHC               52.803
urbanteledensity  38.520
sexratio          38.009
GERupprimary      35.619
govthospital      32.176
highway           28.476
ANC                7.476
electricitycon     0.000
> summary(fit$finalModel)

Call:
lm(formula = .outcome ~ femalelitrate + GERupprimary + govthospital +
    rurexpnd, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-16.1790  -3.6044  -0.0527   4.9559  16.8110

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.216e+01  1.591e+01   5.792 1.73e-05 ***
femalelitrate -3.408e-01  2.341e-01  -1.456  0.16267
GERupprimary  -2.309e-01  1.733e-01  -1.333  0.19929
govthospital   3.832e-05  2.664e-05   1.438  0.16745
rurexpnd      -2.791e-03  8.229e-04  -3.392  0.00325 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.484 on 18 degrees of freedom
Multiple R-squared:  0.6233,    Adjusted R-squared:  0.5396
F-statistic: 7.446 on 4 and 18 DF,  p-value: 0.00101

> fit$finalModel

Call:
lm(formula = .outcome ~ femalelitrate + GERupprimary + govthospital +
    rurexpnd, data = dat)

Coefficients:
  (Intercept)  femalelitrate   GERupprimary   govthospital       rurexpnd
    9.216e+01     -3.408e-01     -2.309e-01      3.832e-05     -2.791e-03

> fit$results
  parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
1      none 14.88681 0.8384254 12.98719 8.694916  0.3118293 7.510255
> model<-fit$finalModel
>
```

Our regression equation is:

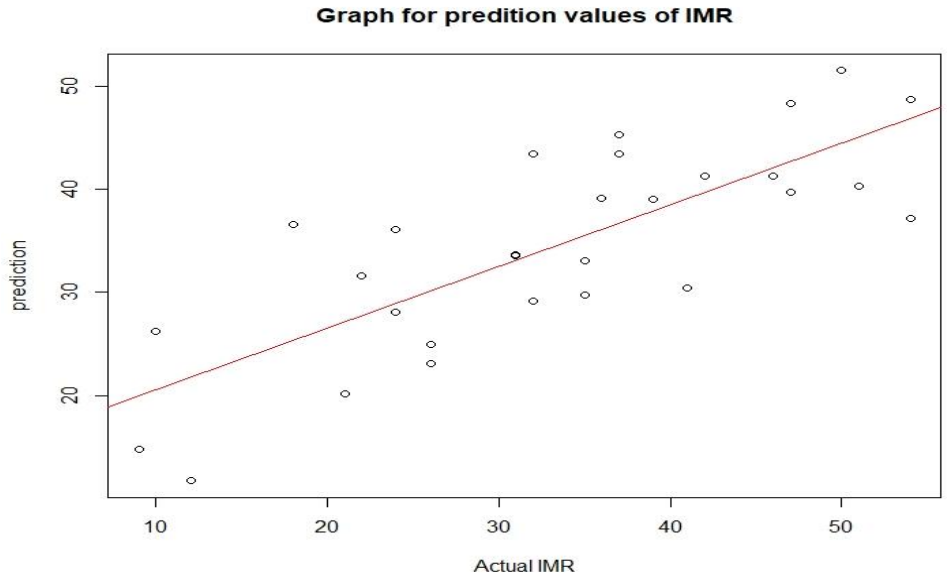
IMR =25.0517-1.25*femalelitrate-0.85*GERupprimary+0.026*govthospital-0.14*rurexpnd.

```
> predict1<- predict(fit$finalModel)
> predict1
      X4      X18      X17      X27      X22      X16       X1       X6      X14      X11      X21
51.55228 45.31384 33.56386 41.29297 30.39141 48.78747 43.42980 31.65128 11.69473 29.79524 43.42431
     X10       X5      X15      X26      X12      X29      X13       X3      X20      X28       X9
40.35760 20.14449 26.17901 29.16357 39.70782 39.05272 28.06299 33.69209 39.14585 37.18901 48.32870
      X2
23.07897
> predict2<- predict(fit$finalModel,newdata = test)
> predict2
       7        8       19       23       24       25
24.95008 36.64494 33.12705 14.74862 36.14109 41.34096
> predict<-c(43.42980, 23.07897,33.69209,51.55228, 20.14449,31.65128,24.95008,36.64494,48.32870,40.357
60,29.79524,39.70782,28.06299,11.69473,26.17901,48.78747,33.56386,45.31384,33.12705,39.14585,43.42431,
30.39141,14.74862,36.14109 ,41.34096,29.16357,41.29297,37.18901,39.05272)
> predict
 [1] 43.42980 23.07897 33.69209 51.55228 20.14449 31.65128 24.95008 36.64494 48.32870 40.35760
[11] 29.79524 39.70782 28.06299 11.69473 26.17901 48.78747 33.56386 45.31384 33.12705 39.14585
[21] 43.42431 30.39141 14.74862 36.14109 41.34096 29.16357 41.29297 37.18901 39.05272
> plot(predict~data2$IMR, xlab="Actual IMR", ylab ="prediction", main="Graph for predition values of I
MR")
> abline(lm(predict~data2$IMR),data=data2, col="RED")
```

Before, describing regression assumptions and regression diagnostics,
we start by explaining two key concepts in regression analysis: Fitted values and residuals errors.
These are important for understanding the diagnostic plots presented hereafter.

Fitted values and residuals

The **fitted** (or **predicted**) values are the y-values that you would expect for the given $x_1, x_2, \ldots, x_6$-values according to the built regression model (or visually, the best-fitting straight regression line).
From the scatter plot below, it can be seen that not all the data points fall exactly on the estimated regression line. This means that, for x values, the observed (or measured) IMR values can be different from the predicted IMR values. The difference is called the **residual errors**.



```
> olsrr::ols_mallows_cp(fit$finalModel, fullmodel2)
[1] 4.325234
```

The number of variables is 4 in our proposed model and the mallow's cp is giving 4.32534 i.e., our model is close enough to the number of variable. Now the adjusted R^2 and multiple R^2 are high enough . So the result is saying that our model is good.

Regression assumptions

Linear regression makes several assumptions about the data, such as:

1. **Linearity of the data**. The relationship between the predictor (x) and the outcome (y) is assumed to be linear.
2. **Normality of residuals**. The residual errors are assumed to be normally distributed.
3. **Homogeneity of residuals variance**. The residuals are assumed to have a constant variance (**homoscedasticity**).
4. **Independence of residuals error terms**.

We will check whether or not these assumptions hold true. Potential problems include:

1. **Non-linearity** of the outcome - predictor relationships
2. **Heteroscedasticity**: Non-constant variance of error terms.
3. **Presence of influential values** in the data that can be:
   - Outliers: extreme values in the outcome (y) variable
   - High-leverage points: extreme values in the predictors (x) variable

All these assumptions and potential problems can be checked by producing some diagnostic plots visualizing the residual errors.

Checking Normality of Residuals:

In R, we can easily augment our data to add fitted values and residuals by using the function `augment()` [broom package]. Let's call the output `model.diag.metrics1` because it contains several metrics useful for regression diagnostics.

```
> model <- lm(IMR~femalelitrate+GERupprimary+govthospital+rurexpnd ,data=data2)
>
> model.diag.metrics <- augment(model)
> ad.test(model.diag.metrics$.resid)

        Anderson-Darling normality test

data:  model.diag.metrics$.resid
A = 0.23504, p-value = 0.7713
```

Here p-value is greater than 0.05. So we can say that , the model residuals follows the normality.

Checking Homoscedasticity:

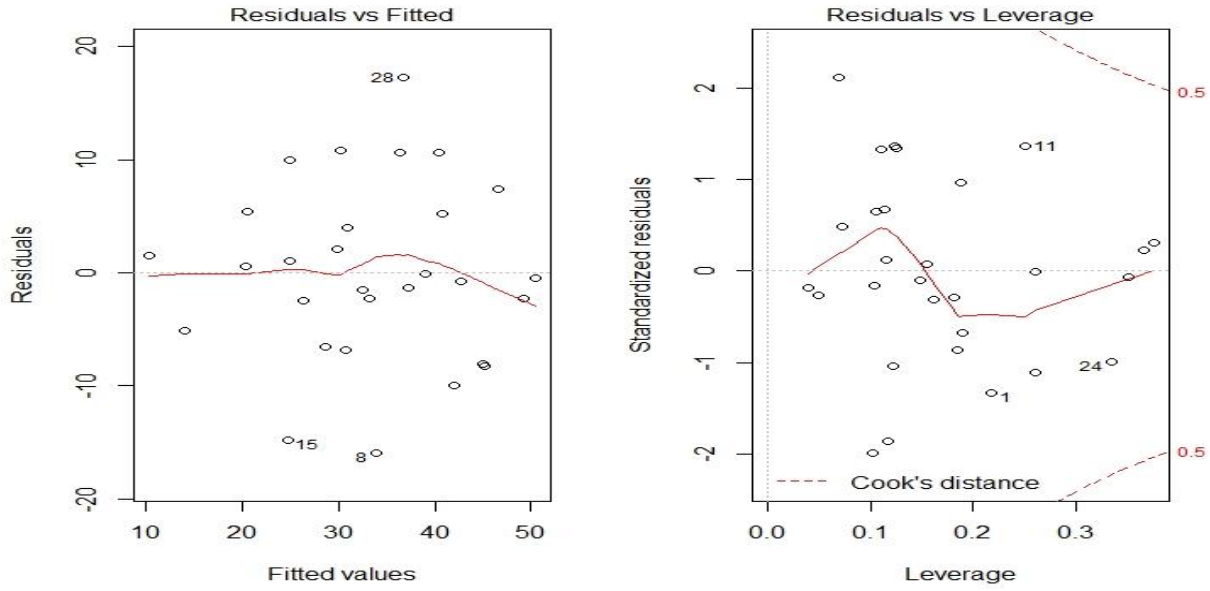
```
> olsrr::ols_test_breusch_pagan(model)

 Breusch Pagan Test for Heteroskedasticity
 -----------------------------------------
 Ho: the variance is constant
 Ha: the variance is not constant

             Data
 -----------------------------
 Response : IMR
 Variables: fitted values of IMR

        Test Summary
 ----------------------------
 DF              =    1
 Chi2            =    0.06833918
 Prob > Chi2     =    0.7937705
>
```

Here P-value is greater than significant level. So we can conclude that here is no presence of heteroscedasticity.

Diagnostic plots

Regression diagnostics plots can be created using the R base function `plot()`

```
> par(mfrow=c(1,2))
> plot(fit$finalModel, 1)
> plot(fit$finalModel, 5)
```



The diagnostic plots show residuals in four different ways:

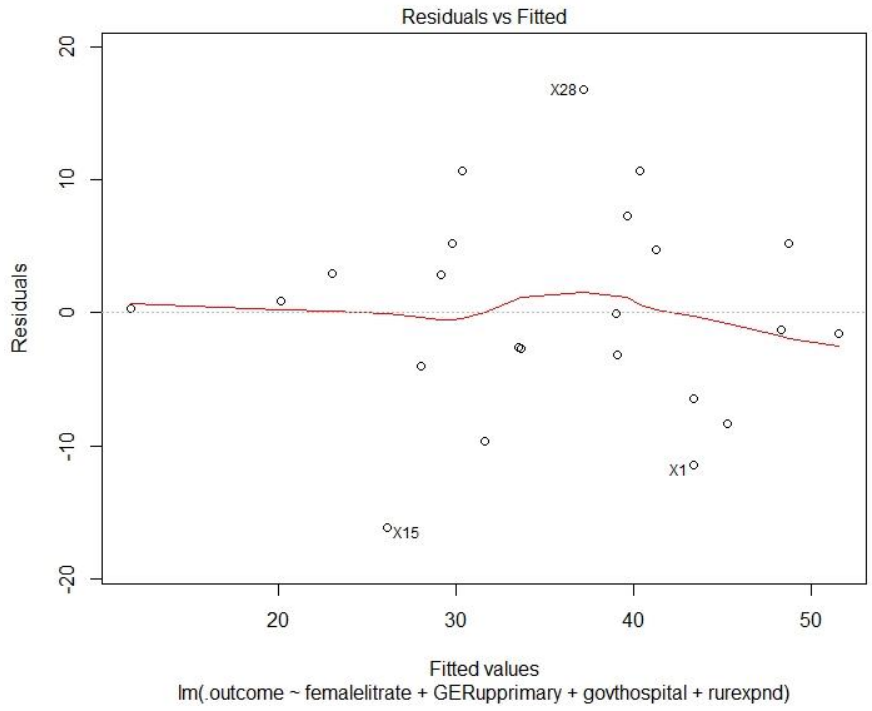
1. **Residuals vs Fitted**. Used to check the linear relationship assumptions. A horizontal line, without distinct patterns is an indication for a linear relationship, what is good.
2. **Normal Q-Q**. Used to examine whether the residuals are normally distributed. It's good if residuals points follow the straight dashed line.
3. **Scale-Location** (or Spread-Location). Used to check the homogeneity of variance of the residuals (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity.
4. **Residuals vs Leverage**. Used to identify influential cases, that is extreme values that might influence the regression results when included or excluded from the analysis. This plot will be described further in the next sections.

The four plots show the top 3 most extreme data points labelled with the row numbers of the data in the data set. They might be potentially problematic. We want to take a close look at them individually to check if there is anything special for the subject or if it could be simply data entry errors. We'll discuss about this in the following sections.

Linearity of the data

The linearity assumption can be checked by inspecting the **Residuals Vs Fitted** plot (1st plot):

```
> plot(model, 1)
```



.

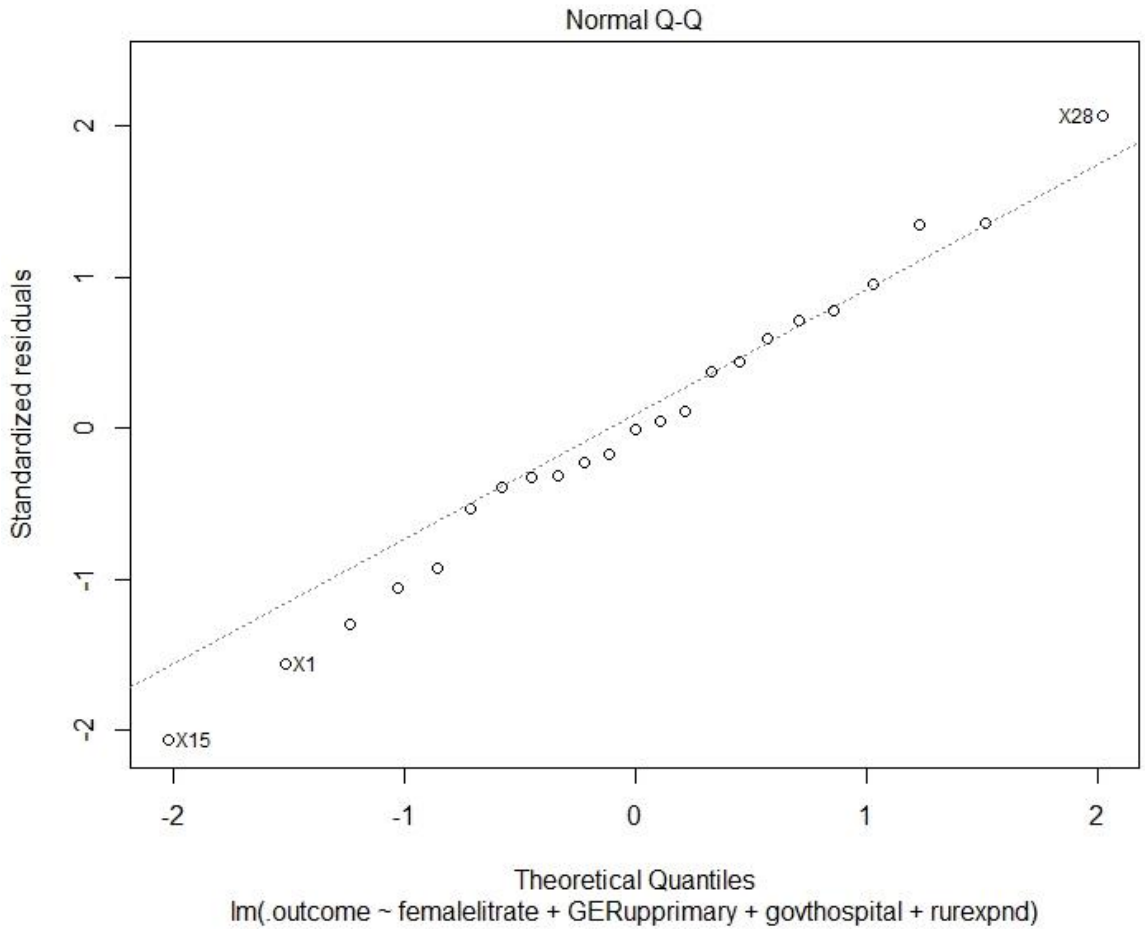
Normality of residuals

The QQ plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line.

In our example, all the points fall approximately along this reference line, so we can assume normality.

```
> plot(model2, 2)
```

In our example, this results are good.

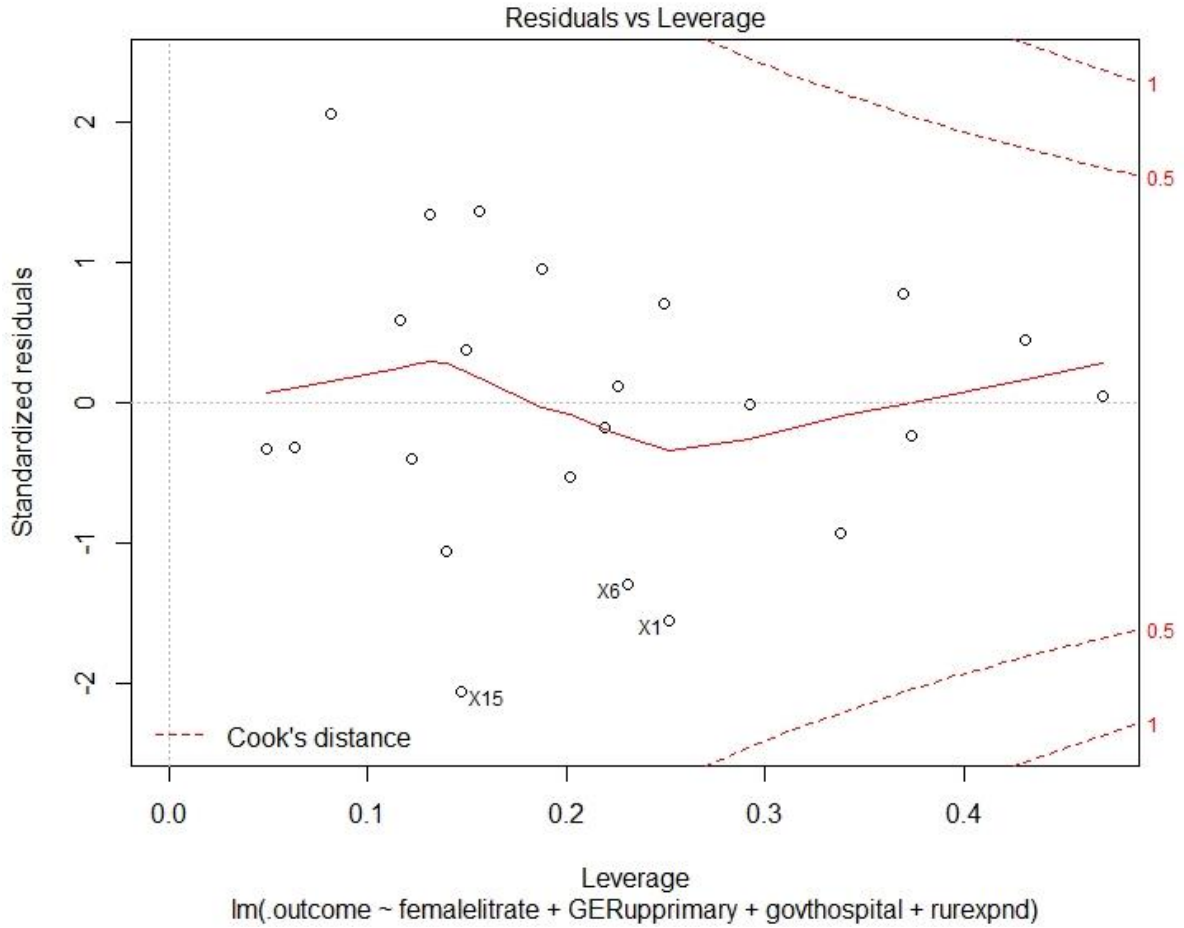


Outliers and high leverage points

Outliers:
An outlier is a point that has an extreme outcome variable value. The presence of outliers may affect the interpretation of the model, because it increases the RSE.

Outliers can be identified by examining the *standardized residual* (or *studentized residual*), which is the residual divided by its estimated standard error. Standardized residuals can be interpreted as the number of standard errors away from the regression line.
Observations whose standardized residuals are greater than 3 in absolute value are possible outliers (James et al. 2014).
High leverage points:
A data point has high leverage, if it has extreme predictor x values. This can be detected by examining the leverage statistic or the *hat-value*. A value of this statistic above `2(p + 1)/n` indicates an observation with high leverage (P. Bruce and Bruce 2017); where, `p` is the number of predictors and `n` is the number of observations. In our example the leverage statistic gives 1.44.
Outliers and high leverage points can be identified by inspecting the *Residuals Vs Leverage* plot:

```
> plot(model, 5)
```



From this plot we can conclude that there is no outliers and high leverage points that can influence the dataset.

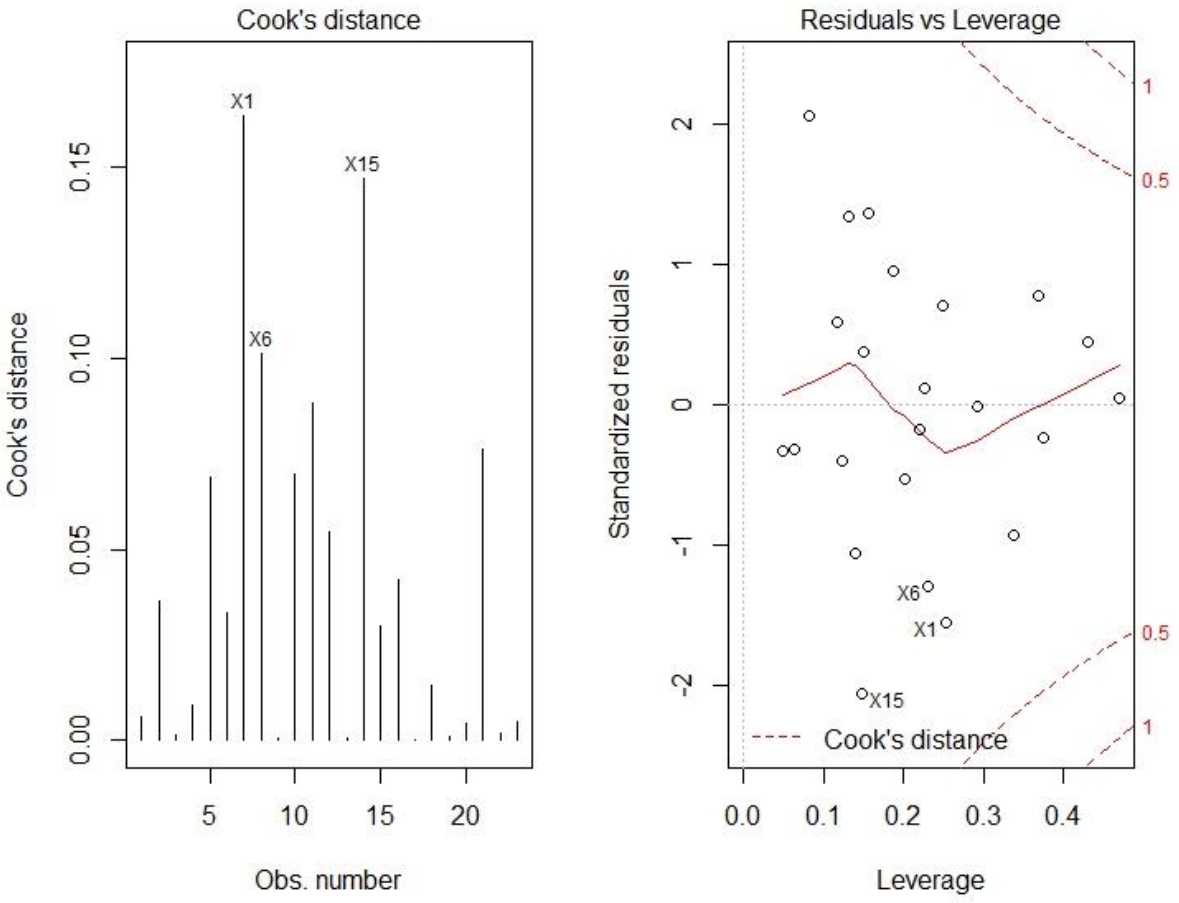
Influential values

An influential value is a value whose inclusion or exclusion can alter the results of the regression analysis. Such a value is associated with a large residual.

Not all outliers (or extreme data points) are influential in linear regression analysis.

Statisticians have developed a metric called *Cook's distance* to determine the influence of a value. This metric defines influence as a combination of leverage and residual size.
A rule of thumb is that an observation has high influence if Cook's distance exceeds `p/(n - p - 1)` (P. Bruce and Bruce 2017), where `n` is the number of observations and `p` the number of predictor variables.
The *Residuals Vs Leverage* plot can help us to find influential observations if any. On this plot, outlying values are generally located at the upper right corner or at the lower right corner. Those spots are the places where data points can be influential against a regression line.
The following plots illustrate the Cook's distance and the leverage of our model:

```
> mycd<-cooks.distance(model)
> mycd1<-round(mycd, 4)
> sort(mycd1)
   X29    X14     X5     X3    X17     X9    X20     X2     X4    X27    X13    X26    X16    X18
0.0000 0.0004 0.0008 0.0011 0.0013 0.0018 0.0044 0.0049 0.0064 0.0092 0.0145 0.0298 0.0334 0.0362
   X12    X10    X22    X11    X28    X21     X6    X15     X1
0.0420 0.0545 0.0688 0.0699 0.0763 0.0884 0.1014 0.1473 0.1636
> par(mfrow= c(1,2))
> plot(model, 4)
> plot(model, 5)
```



By default, the top 3 most extreme values are labelled on the Cook's distance plot. They are 15, 1 and 6.

The equation of 1st model is:

IMR =25.0517-1.25*femalelitrate-0.85*GERupprimary+0.026*govthospital-0.14*rurexpnd.

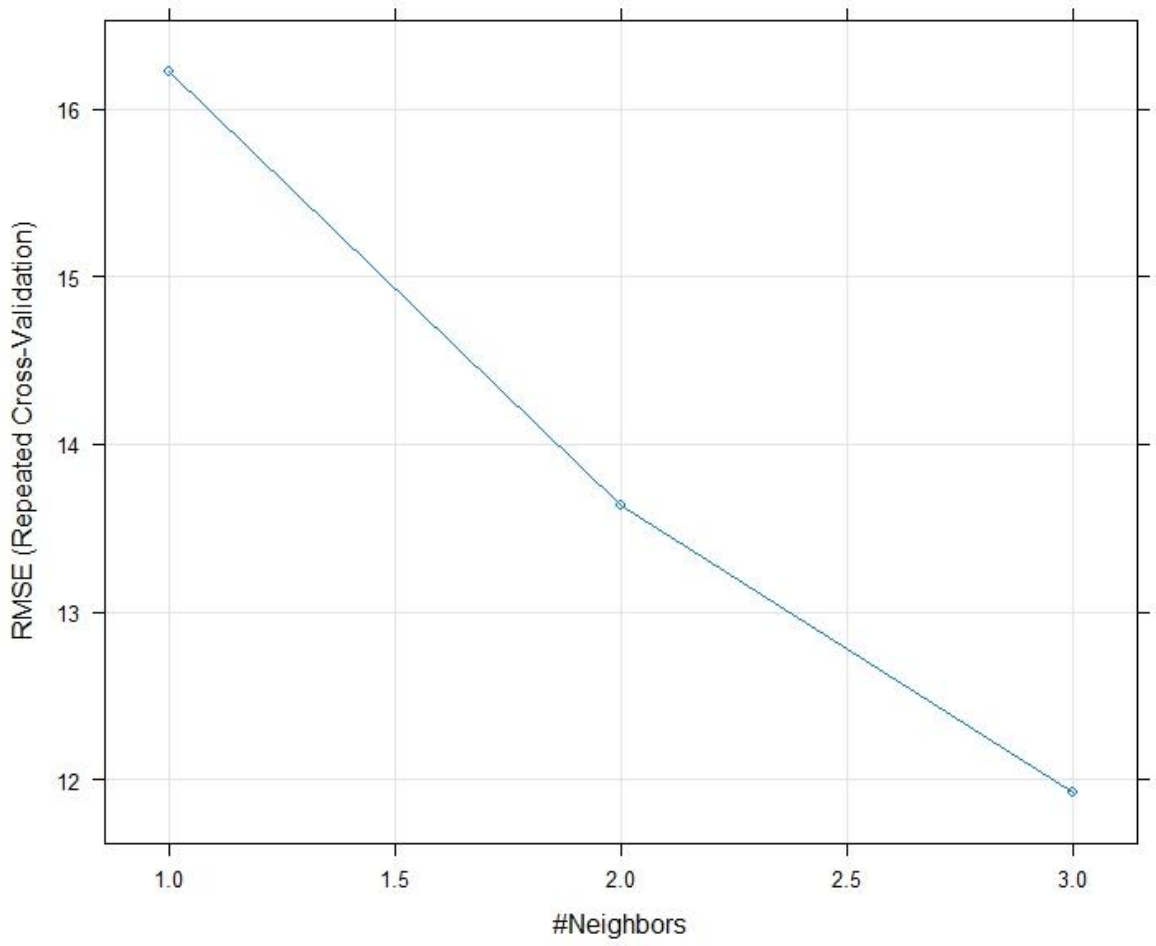
k-Nearest Neighbor Model

```
> set.seed(8557)
> ranknn<-sample(1:nrow(data2),0.8*nrow(data2))
> ranknn
 [1]  3 27 13 29 22 11 23  8 28 15  1  5 16  2 14 18 20 17 24 25  9 12  7
> trainingknn <- data2[ranknn,]
> testknn <- data2[-ranknn,]
> View(trainingknn)
> #trcontrol<-trainControl(method = 'oob')
> trcontrolknn<-trainControl(method = 'repeatedcv',number = 10,repeats = 1
000)
> set.seed(533)
> fitknn <- train(IMR ~.,data = trainingknn, tuneGrid= expand.grid(k=1:3),
method = 'knn',trControl=trcontrolknn, preProc=c('center','scale'))
```

```
> fitknn
k-Nearest Neighbors

23 samples
11 predictors

Pre-processing: centered (11), scaled (11)
Resampling: Cross-Validated (10 fold, repeated 1000 times)
Summary of sample sizes: 21, 20, 21, 21, 21, 21, ...
Resampling results across tuning parameters:

  k  RMSE      Rsquared   MAE
  1  16.22587  0.8363818  14.82833
  2  13.63659  0.8334974  12.53093
  3  11.92114  0.8463211  10.91330

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 3.
>
```

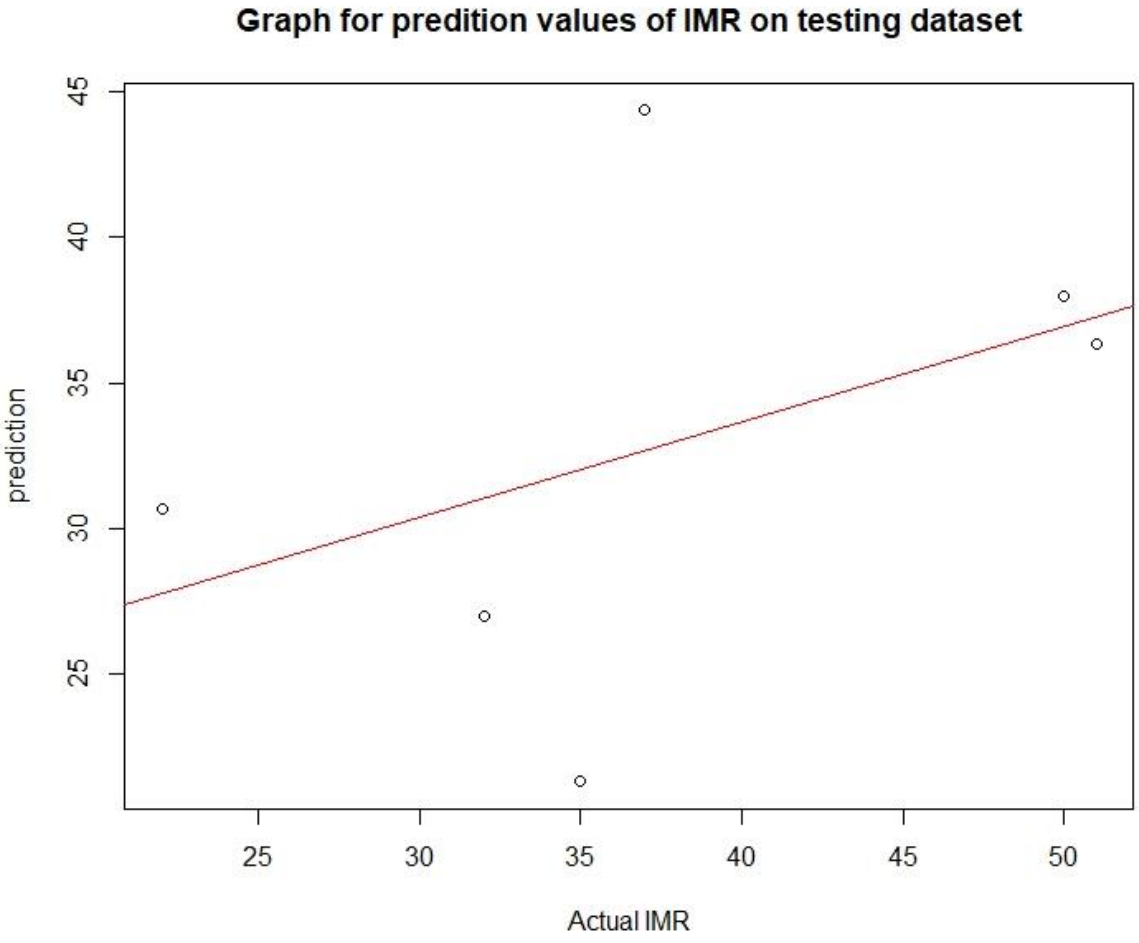
R-Squared Value of our 2nd model is 0.8463211

```
> plot(fitknn)
```




```
> varImp(fitknn)
loess r-squared variable importance

                 Overall
femalelitrate    100.000
highway           77.392
ruralteledensity  75.836
rurexpnd          67.859
CHC               48.519
electricitycon    36.826
urbanteledensity  22.185
ANC                6.917
GERupprimary       5.408
sexratio           2.889
govthospital       0.000
```

```
> fitknn$finalModel
3-nearest neighbor regression model
> predictknn<- predict(fitknn,newdata = testknn)
> predictknn
[1] 38.00000 30.66667 36.33333 21.33333 44.33333 27.00000
> RMSE(predictknn,testknn$IMR)
[1] 10.79952
> par(mfrow=c(1,2))
> plot(predictknn~testknn$IMR, xlab="Actual IMR", ylab ="prediction",
 main="Graph for predition values of IMR")
> abline(lm(predictknn~testknn$IMR),data=data2, col="RED")
```



Random Forest Model:

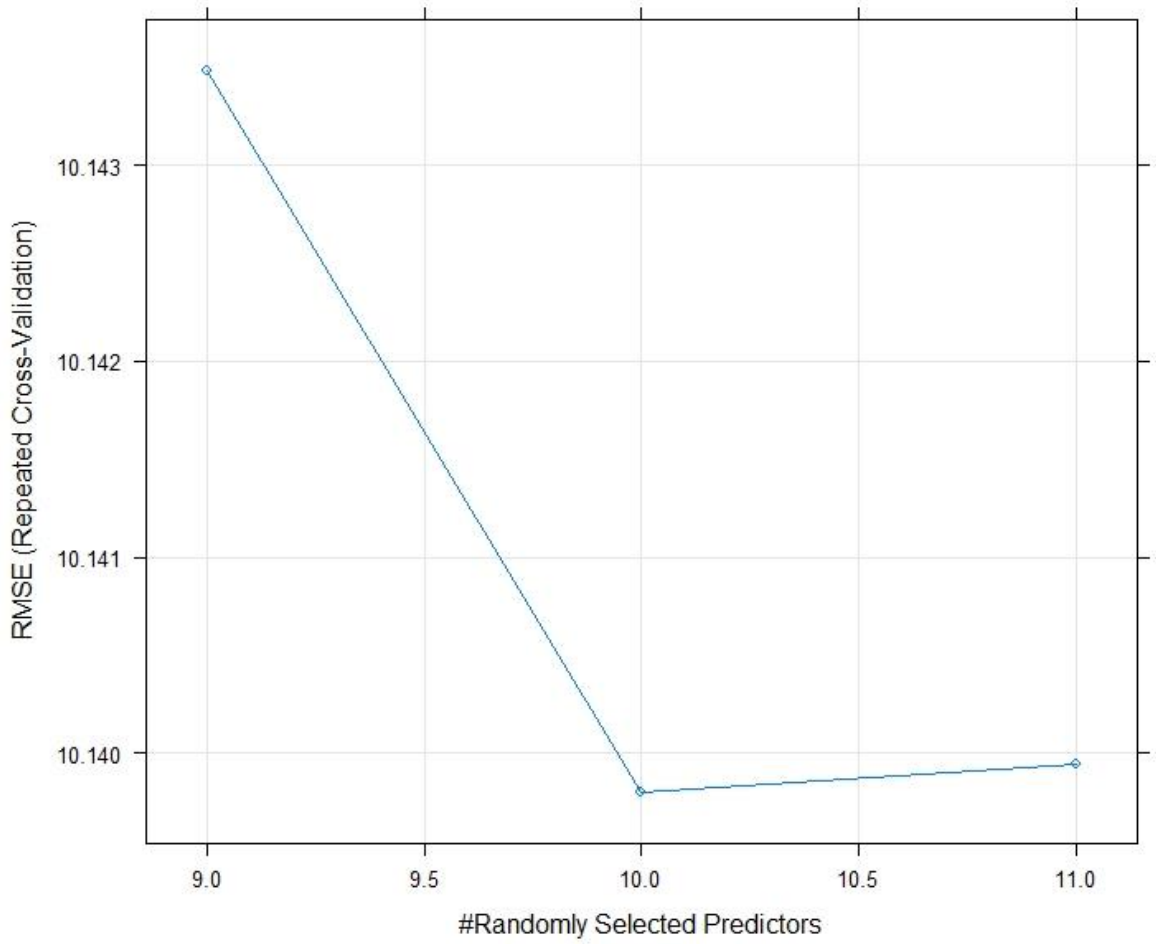
```
> set.seed(7954)
> ranrf<-sample(1:nrow(data2),0.8*nrow(data2))
> ranrf
 [1] 24 23  8  6 14  5  1 10  7 27 21 15  4 11 13 19 26  3 28  9 20 22 16
> trainingrf <- data2[ranrf,]
> testrf <- data2[-ranrf,]
```

```
> View(trainingrf)
> #trcontrolrf<-trainControl(method = 'oob')
> trcontrolrf<-trainControl(method = 'repeatedcv', number = 10, repeats =
1000,search = "random")
> set.seed(257)
> tunegrid<-expand.grid(.mtry=c(1:15))
> fitrf <- train(IMR ~.,data = training, tunegrid = tunegrid ,method = 'rf
', trControl=trcontrol, ntree = 1000)
```

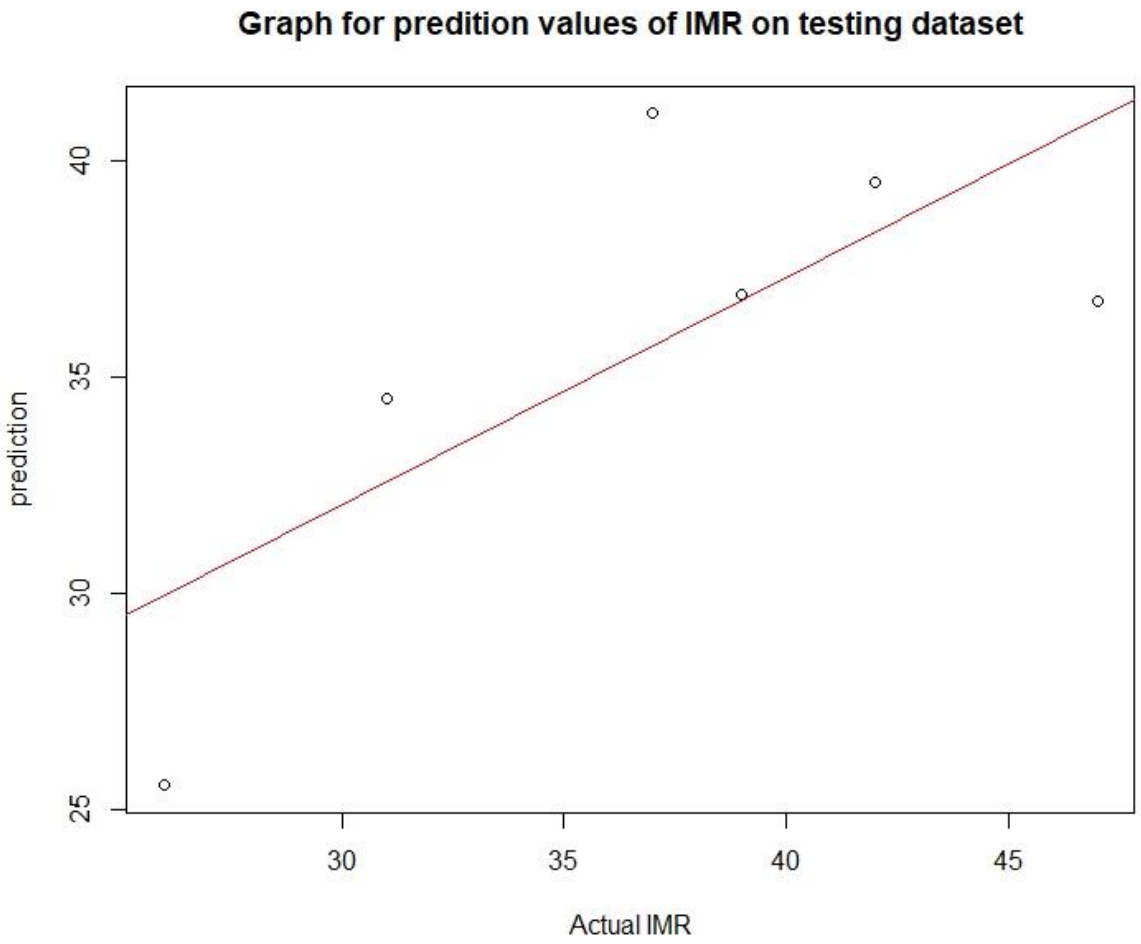
```
> fitrf
Random Forest

23 samples
11 predictors

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 1000 times)
Summary of sample sizes: 21, 20, 21, 21, 21, 21, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   9    10.14348  0.8717020  9.234327
  10    10.13980  0.8714829  9.220014
  11    10.13994  0.8716231  9.203547

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 10.
>
```

R-Squared of our 3rd model is 0.8714829.

```
> plot(fitrf)
```




```
> varImp(fitrf)
rf variable importance

                  Overall
femalelitrate    100.0000
rurexpnd          43.0391
sexratio          21.9323
CHC               15.0575
ruralteledensity   9.5959
electricitycon     5.6841
GERupprimary       3.5112
highway            1.9854
govthospital       0.9799
ANC                0.8184
urbanteledensity   0.0000

> predictrf<- predict(fitrf,newdata = testrf)
> predictrf
       2       12       17       18       25       29
25.57218 36.76133 34.53140 41.10577 39.51907 36.91458
> RMSE(predictrf,testrf$IMR)
[1] 4.913339
> #plot(fitrf, main="Graph for repeatedcv vs forest of IMR")
> plot(predictrf~testrf$IMR, xlab="Actual IMR", ylab ="prediction", main="
Graph for predition values of IMR on testing dataset")
> abline(lm(predictrf~testrf$IMR),data=A, col="RED")
```



Result

After comparing the 1st, 2nd , 3rd model we can conclude that all of these models are giving almost same R-squared values. So, we can conclude that, our 1st model or the primary model (generated from stepwise regression technique):

IMR =25.0517-1.25*femalelitrate-0.85*GERupprimary+0.026*govthospital-0.14*rurexpnd.

Can be used as a suitable model for our IMR data set.

This result of our primary model gives us:

- Linearity in the data.
- Normality in the data.
- Homoscedasticity
- No outliers in the data set.

- The values of multiple R^2 are very high.
- The value of mallow's cp is very close to the number of variable in the datasets.
- The proposed model is the suitable model for our dataset.

The model is:

IMR ~ femalelitrate + GERupprimary + govthospital + rurexpnd