Linear Regression

what is predictive modelling?

If somebody asks you to guess how Virat Kohli is going to perform in next cricket match against Australia, you can take a guess *[Assuming you are cricket follower]*. This guess is based on your knowledge of his past performances in these or similar conditions and against this kind of team.

Predictive modelling gives this process a formal framework. It gives you tools to extract mathematical equations/rules from the past data to predict future results.

Steps of Predictive Modelling: 1. After identifying the Business objectives, first step in any predictive model is to collate data from various sources. The sources of data can be historical data, demographic data, behavioural data, Customer data and transactions data etc. 2. In the second step, we need to prepare data into right format for analysis. Here we normally clean data, impute missing values, transform and append variables. 3. Based on the Business Objectives we have to select on or combination of Modelling Techniques like Linear regression for predicting the future Values. 4. Final step is to check the performance of Model like Error, accuracy, ROC and other measures.

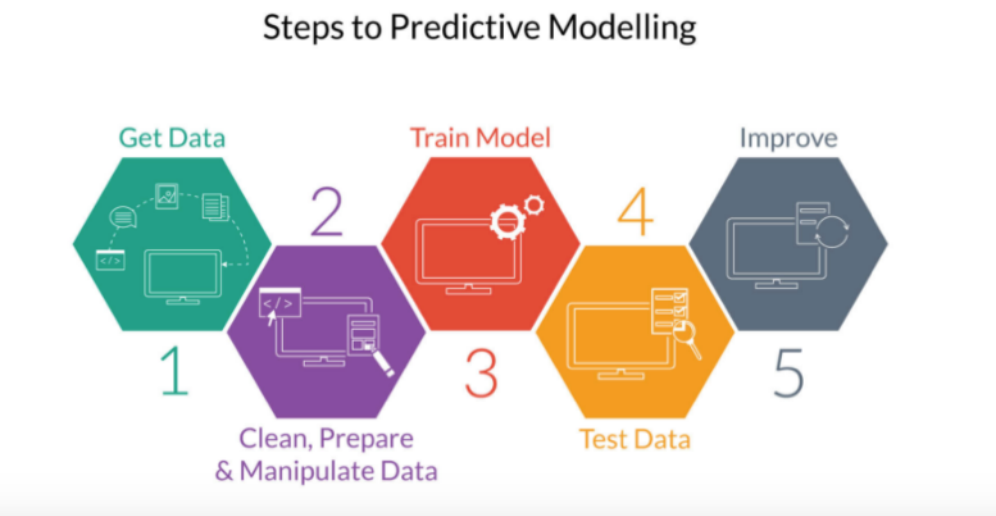


Figure 1: Steps of Predictive Modelling

Before you draw any conclusions about **predictive modeling** being some kind of black magic, lets list down its limitations:

1. The models *[equations/rules]* are dependent on the past data that you have. If data is bad , your predictive models are also going to be bad.
2. Every model will have errors associated with its predictions. Better the model, lesser the error, but it will never be an exact estimate for all practical purposes.
3. Model will be good only until underlying factors on which it was based on , do not change behavior. For example: A predictive model which was built to predict a particular share performance in good economic conditions will perform rather poorly in recession.

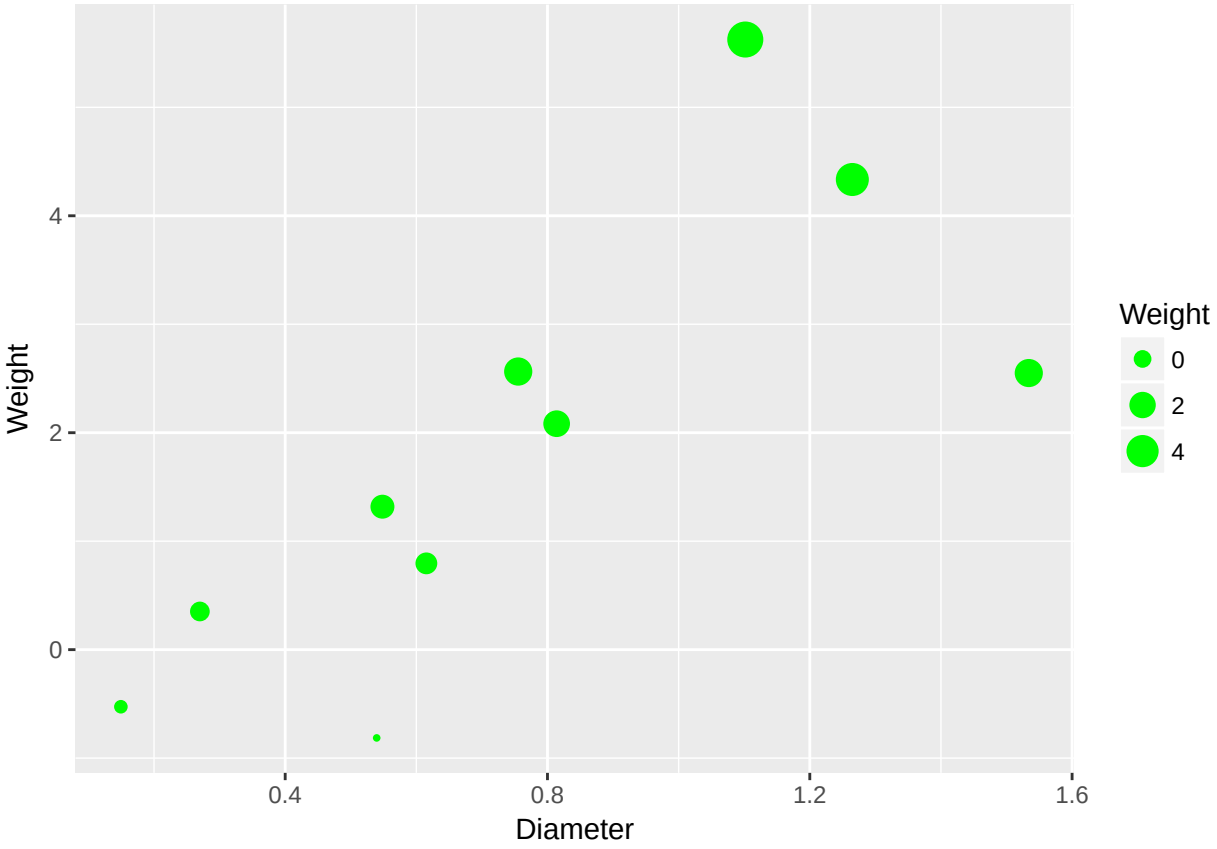
Before we right away jump in to predictive modelling and start extracting those said equations, lets first figure out what really leads us there:

Correlation

When we say that, given past data we can predict future results/outcomes, we are essentially relying on our hunch that we can observe some other factor which affects the outcome and we can leverage that information. For example: when you pick up an apple and guess it weight, you are betting on your assumption that weight of an apple is dependent on its size *[or diameter]*.

In other words , you are assuming that weight of that apple is **correlated** with its diameter.

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```



As you can see in the figure, you were not really wrong. To make this more concrete we need to figure out a way to quantify this *correaltion* . Before doing that, we also need to understand different kind of correlations. As observed in the figure above, weight of apple goes up as its diameter increases. On looking more closely you find out that, increase in weight of apples is happening in *possibly* constant multiples of increase in diameter. This is called **linear correlation**. There can be other form of correlations as well. (Figure 1)

What do we mean by these *linear* and *other* forms of correlations is that one variable *[lets say y]* can be written as a function of another *[lets say x]*

- Linear Correlation : y = ax + b
- Exponential Correlation : y= a*exp(x) + b

quantifying correlation coefficient

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

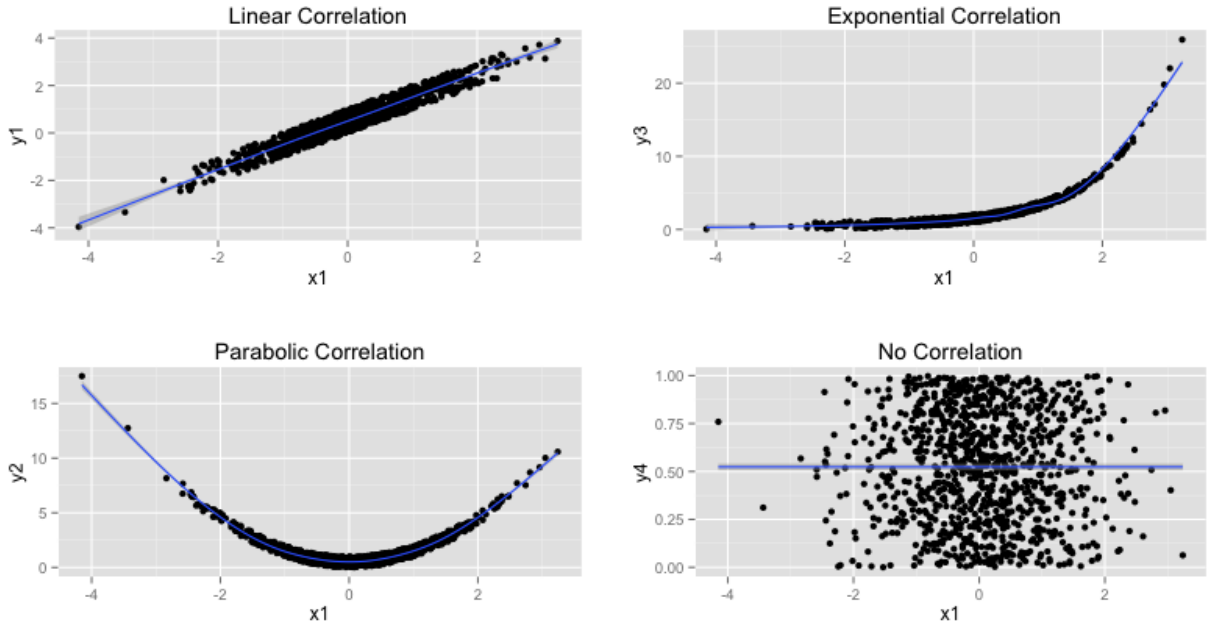


Figure 2: Linear and Non-Linear Correlation

This formula is designed to measure strength of **linear** correlation. it has following properties:

- -1 < r < 1
- It takes -ve values of negative correlation and +ve values of +ve correlation *Note: -ve correlation between x and y means , when x increases y decreases and vice versa*
- Correlation is strong when absolute value of r is close to 1, it is weak if the absolute value is close to zero
- Value of r doesn't change if you linearly transform any or both of the variables. Meaning, correlation between x and y will be same as correlation between (ax+b) and (Ay+C).
- As emphasized above , it can be used to measure linear correlation only

But this formula being capable of measuring only linear correlation is not limited. look at this equation again:

- y= a*e^x + b

consider that x' = e^x . You can write the above as this :

- y= a*x' + b

which is same as saying that y and x' are linearly correlated. In a similar manner, if you observe that y and x are non-linearly related , you can apply a suitable transformation to either of x and y and make the relationship linear.You can then measure the strength of correlation between y and x' *[transformed variable]*

Correlation and Causation (Figure 2)

Causation is when a particular factor is the reason for change in another factor. For example number of people buying sun-screen in the city and city's temperature are going to be correlated. Also there is direct causation. Temperatures going up *[Hot Sunny Weather]* is driving sales of sun-screen. However if two factors are correlated , that doesn't guarantee that there will be causation. For example look at following figure :

Figure 3: Correlation and Causation

Clearly you can see that ice-cream sales and shark attacks are correlated as far as the data can be seen. However that doesn't mean that ice-cream sales are cause of shark attacks. In fact rising temperature cause more people to buy ice-cream and also it causes people to go to beaches in larger numbers *[and sometimes subsequently get attacked by sharks]* . This clarifies two things:

- Clearly correlation doesn't necessarily means causation
- However it does indicate that correlated factors might have a common underlying cause *[Again, not always neccessary]*

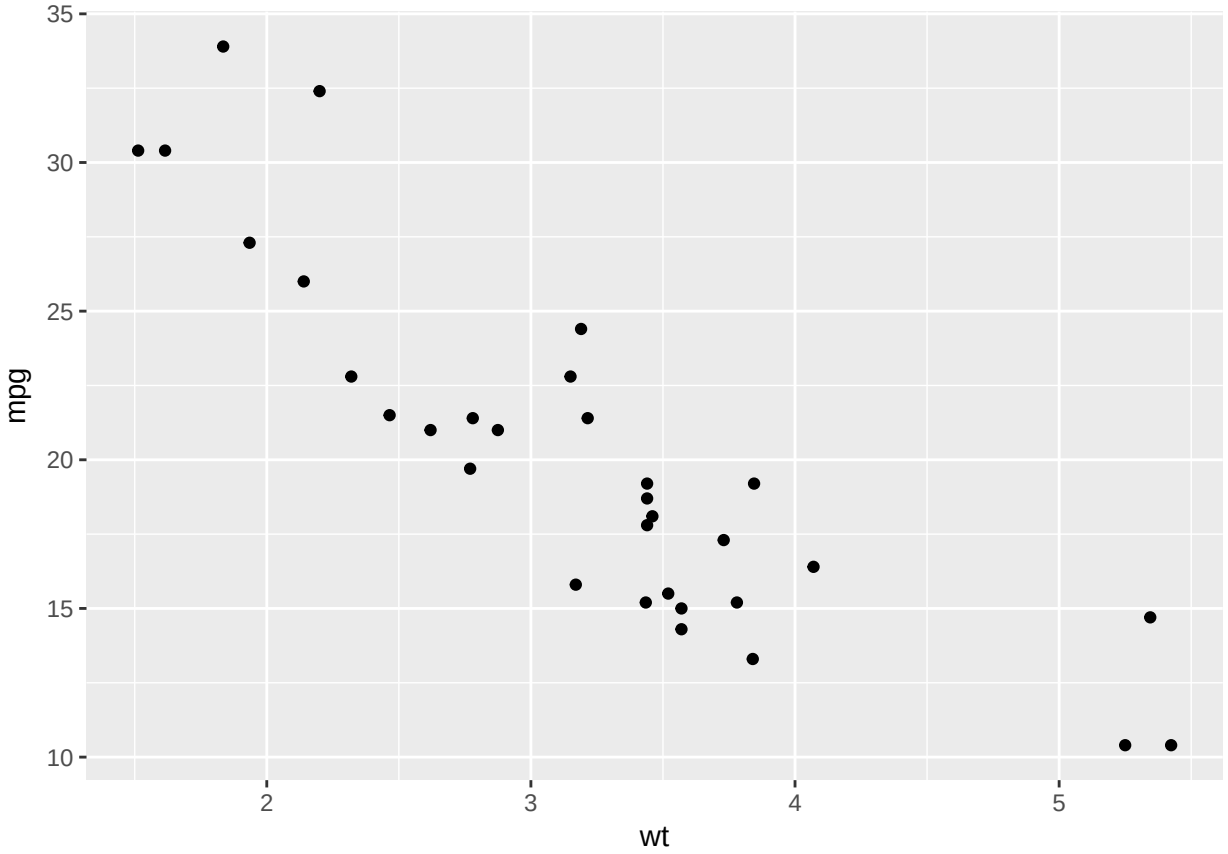
Finding Correlation in R : Correlation Coefficient and Scatter Plots

```
cor.test(mtcars$mpg,mtcars$wt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$mpg and mtcars$wt
## t = -9.559, df = 30, p-value = 1.294e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594
```

This tells you that correlation exists and it is significant. Also the sign of correlation coefficient is -ve. Which tells you that as weight of a vehicle increases , mileage goes down. Lets look at that visuall with the help of a scatter plot.

```
ggplot(mtcars,aes(x=wt,y=mpg))+geom_point()
```



This tells you the same story, although without numbers.

Simple Linear Regression

Well, correlation coefficient lets us figure whether a pair of variables are affecting each other and if they are then how strong is that effect. However , what we eventually want is an equation which can be used to predict value of y [my response/target/outcome], given a value of x [my input/predictor].

Formally Linear regression attempts to fit a linear relation between a variable of interest (response variable) and a set of predictor variables that may be related to the variable of interest.

As we have seen earlier , if two variables are linearly correlated we can essential draw line through their scatter plot which depicts the relationship between them. But the problem is we can draw many lines, and until now have no clue as to which one to chose finally. (Figure 3)

However , you can observe few crucial things here :

- It's impossible to come up with a line which passes through all the points, in other words ; whatever line equation you come up with, there are going to be errors associated with it.
- Take general equation of a line ; $y = \beta_0 + \beta_1 * x$, what is changing between these lines in the figure is the value of the parameters b0 and b1 . We need to find out such values of these parameters for which error is minimum.

In the figure below, red points on the line are your predictions for values of y given some values of x, whereas blue points are the actuals observed for those values of x. (Figure 4)

After looking at the figure above and the crucial observations that we mentioned, you must have realized that actual value of y can be written as addition of predicted and associated error.

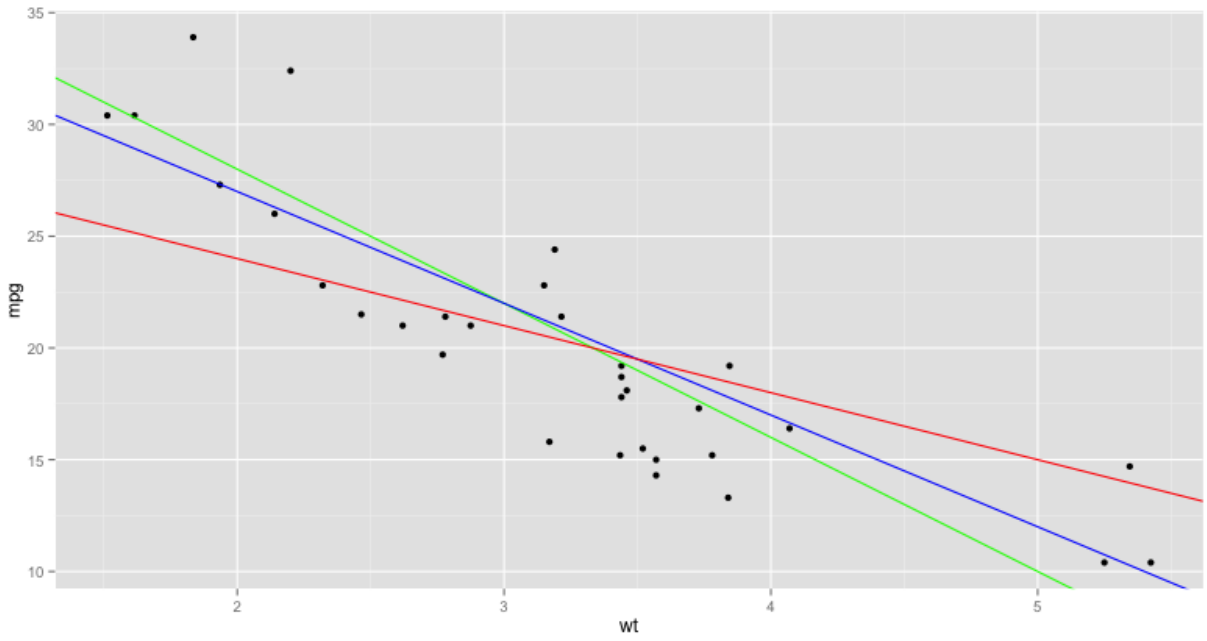


Figure 4: Which Lines to Pick

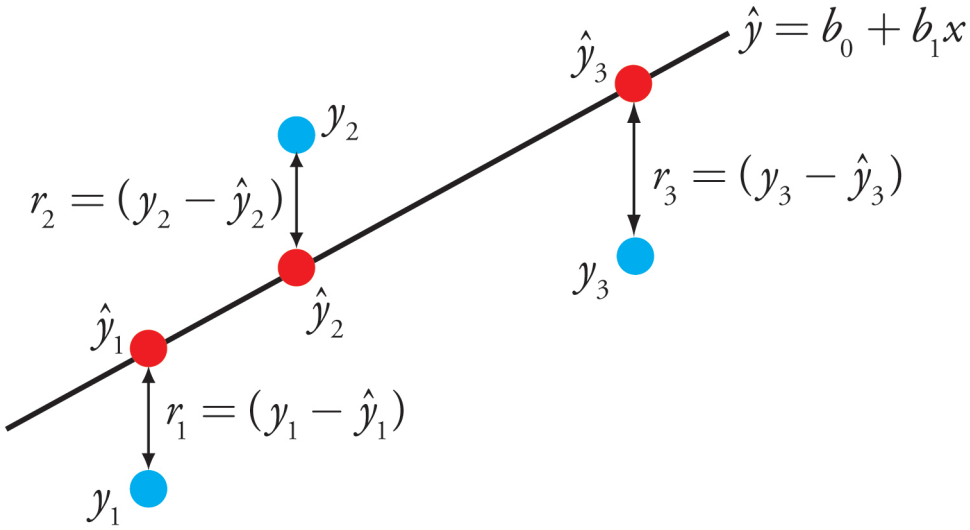


Figure 5: Errors in Predictions

$$y_i = \hat{y}_i + e_i$$

$$y_i = \beta_0 + \beta_1 * x_i + e_i$$

from here we can see that:

$$e_i = y_i - \beta_0 - \beta_1 * x_i$$

What we need to minimize is the collective error for entire data.

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 * x_i)^2$$

Now if we want to minimize the above quantity w.r.t. β_0 and β_1 then we can differentiate that equation and put it equal to zero to find values of β_0 and β_1 for which $\sum_{i=1}^{n} e_i^2$ is minimum. Resulting equations are also called normal equations. Here they are:

$$-2 * \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 * x_i) = 0$$

and

$$-2 * \sum_{i=1}^{n} x_i * (y_i - \beta_0 - \beta_1 * x_i) = 0$$

upon solving them you'll get result for β_0 and β_1 which can also be written like this:

$$\beta_1 = r_{xy} * (s_x / s_y)$$

and

$$\beta_0 = \bar{y} - \beta_1 * \bar{x}$$

Don't get intimidated by all these equations, at no stage as a business analyst you'll need to create or solve these equations.Software *[SAS or R or python]* will do this for you. Don't worry and read on.

So lets say your software gave you the appropriate values of β_0 & β_1 and you have your predictive model equation ready. But think, that whatever variable data you pass to this mathematical framework you'll get some values of β_0 & β_1, does that ensure you have a **good** model? Not necessarily .

Imagine , in absence of any such equation, what is your best guess for y . Its $\bar{y}$ the average value. But with this guess there is error associated with each observation : $y_i - \bar{y}$. Writing this in collective form :

SST= Total sum of squares = Total Variability = $\sum_{i=1}^{n} (y_i - \bar{y})^2$

This is also called total variability in target. We intend to bring this down with our model. In other words, we want to explain this variability with our model. Lets try to understand this with this figure:(Figure 6)

Consider that our predicted values are $\hat{y}_i$. As discussed earlier , they don't explain entire variability in the target but a part instead. which is $\hat{y}_i - \bar{y}$.

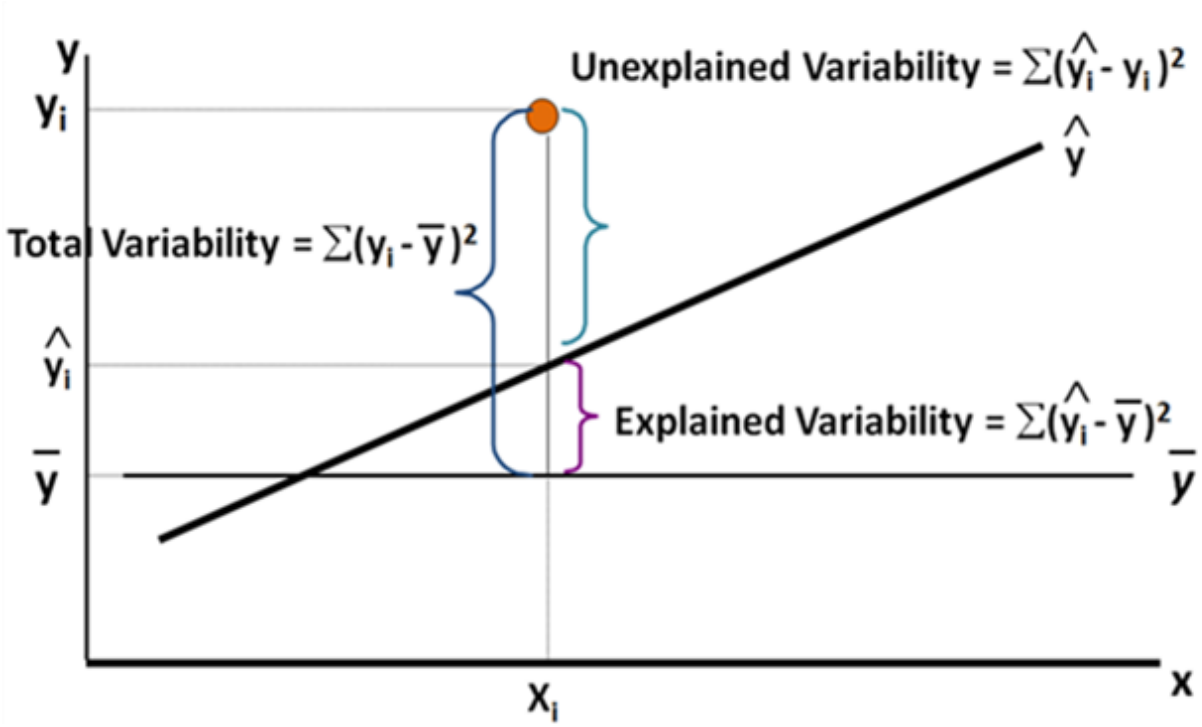


Figure 6: R square

SSR = Regression sum of squares = Explained Variability = $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$

You never get a perfect model. After prediction, each y_i has still error in prediction being equal to $y_i - \hat{y}_i$.

SSE = Error sum of squares = Unexplained Variability = $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Ideally you'd want explained variability to be equal to total variability. That'll be a perfect model. Ratio between explained and total variability is defined as coefficient of determination, it is written as R^2. Its an indicator of how good your model is.

$$R^2 = SSR/SST$$

As you can see R^2's maximum value is 1 when SSR=SST . Also its minimum value would be 0 when SSR=0 . The closer to 1 , R^2's value is, better is your model.

OK, so now we can estimate parameters for our linear model equation, also we can measure how good our model is, but we haven't done anything about errors. Of course there is no way to predict them *[That's why they are errors!]*

What we can do is to make some kind of probabilistic estimate for them.

Interpreting the Estimated parameters of Regression Model

When the predictor variable x_i in our simple linear regression model ($y = \beta_0 + \beta_1 * x$) increases by one unit then the response variable y increases by $beta_1$ units, remaining all other variables are constant.

β_0 represents the intercept term, it simply tells if the x_i value is zero then the y value will be equal to β_0

Assumption of Normality and its consequences

If we assume that $e_i \sim N(0, \sigma_i^2)$, then for each i_{th} Observation we can come up with some confidence interval around our prediction. For example we can say that each i_{th} response would lie in $y_i \pm 3 * \sigma_i$ with 99.7% confidence.

In addition to this quantification of errors , normality assumption allows us to build a very useful hypothesis. Before we discuss that, lets answer a question. We have seen this that if we supply some random y and x variable to this mathematical frame work, we'd get some non-zero estimate of β_0 & β_1 . Does that mean that y *[our response/target/DV]* really depends on x and it follows the equation $y = \beta_0 + \beta_1 * x$?

If y did not depend on x then estimate for β_1 should be zero. But we rarely see that with real data. Estimate might be close to zero but never exactly zero. So when should we conclude that parameter estimate for our β is significantly different from zero?

Now lets look at the hypothesis that the normality assumption enables us to build.

$$H_0 : \beta_i = 0$$
$$test - statistic : \frac{\beta_i}{S_{\beta_i}} \sim t - distribution$$

We can look at the p-values for the test, and if they turn out to be greater than alpha *[standard value 0.05]* , we can conclude that null hypothesis is true and parameter estimate is non-zero by chance and should be discarded.

Assumption of Homoscedasticity

We can estimate error variance σ_i for each x_i from our past data, because we very likely will have multiple observation for x_i. But we are building this model to predict values of y for not yet seen values of x, for such values of x there is no way for us to estimate error variance. We'll have to assume that error variance remains constant across all values of x . This assumption of constant variance across all values of predictor variable *[x]* is called homoscedasticity assumption.

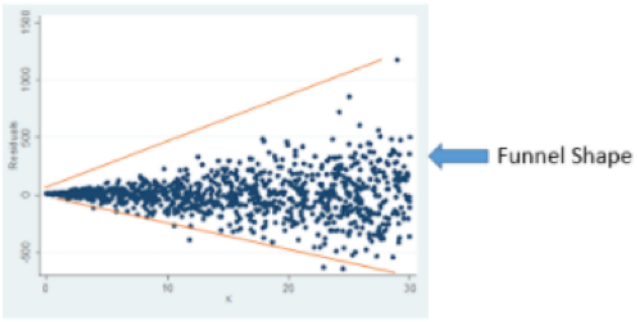


Figure 7: R square

Assumption of error independence

Errors from a model are those part of our response which we could not predict. Whatever part fell in a pattern became part of our model. Errors on the other hand are random and don't follow a pattern. If you plotted error with your target you should ideally see that there is no apparent pattern.

How to check validity of assumption and cosequences of violation

1. Normality Assumption: If errors don't follow normal distribution, we can not rely on p-values and confidence intervals for our predictions. However point estimates for target remain unaffected apart from the fact that parameter estimates might be non-zero by chance. We can check whether this is true or not by looking at qqplot for errors, histogram with kernel density curves for errors.
2. Homoscedasticity Assumption: If you plot errors with target and instead of a random cloud you see a funnel shape or some other pattern , that'd be an indication of hetroscedastic errors. One popular remedy for the same is to take log of response and use that instead. Taking log brings down scale of errors and of course scale of difference between them as well.
3. Independence Assumption: If error seem to follow some pattern with any of the predictor that indicates at non-linear relation between and y and that predictor. We should try appropriate variable transformation instead of using that variable as it is.

Multiple Linear Regression

Multiple linear regression is just an extension of simple linear regression. Although few additional issues come up due to extra variables but basic frame work remains same. Instead of one predictor variable x, you have multiple predictor variables $x_1, x_2, x_3.....x_p$. The target y can still be written as linear combination of these predictors:

$$y_i = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3..... + \beta_p * x_p + e_i$$

Again for finding best values of parameters, we'll minimize error sum of squares $\sum_{i=1}^{n} e_i^2$. Instead of 2 linear equations, now we'll have (p+1) linear equations to solve. Other than that everything else remains pretty much same.

Multi-collinearity

multi-collinearity means, one or more of the predictor variables being linearly dependent on few other predictor variables. There is a very important consequence to this in MLR. Recall the test statistic for hypothesis which we built to figure whether parameter estimate for β_i is significantly different from zero. If that test statistic is close to zero, p-value becomes high and you discard the corresponding predictor variable.

This reliance on p-value to discard not-so-good variable is alright until multiple predictors came into picture and with them , came multicollinearity. Discussion, which is about to follow is a little technical, if you find it overwhelming, don't worry, in the end we'll walk you through a case study done in SAS which you'll be able to understand even if this discussion doesn't make sense to you.

Lets say you have a target y which you are trying model as linear combination of $x_1, x_2, x_3.....x_p$. Forget about y for a minute. Consider that we choose one of the predictor as target and rest as predictor for it. Let x_j is my target and $x_1, x_2, x_3...x_{j-1}, x_{j+1}.....x_p$ are predictors for it. I go on and build a linear regression model and find its coefficient of determination to be R_j^2. if R_j^2 is high that means x_j can be written as linear combination of $x_1, x_2, x_3...x_{j-1}, x_{j+1}.....x_p$. This also implies that x_j contains redundant information. Another repercussion of this comes from the test statistic of hypothesis $\beta_j = 0$.

Remember the test statistic $\frac{\beta_j}{S_{\beta_j}}$. The denominator is proportional to $\frac{1}{(1-R_j^2)}$. As R_j^2 goes close to 1, S_{β_j} goes close to infinity, which means test statistic $\frac{\beta_j}{S_{\beta_j}}$ goes close to zero. Which in turn makes p-values artificially high. And you might end up discarding the variable. Remember that, p-values here are going up, **not** because variable is not correlated enough to y, but because of multicollinearity present in the data.

Which implies if multicollinearity is present in the data, you can not rely on the p-values. We can measure multicollinarity effect caused by presence of each variable by looking at the corresponding $\frac{1}{(1-R_j^2)}$ values. This is also called Variance Inflation Factor because it is the factor by which variance of parameters estimate S_{β_j} goes up.

$$VIF = \frac{1}{(1 - R_j^2)}$$

We will first need to remove variables from our consideration which have high VIF values and then proceed to build our regression model.

Using Categorical Variables

The process that you have seen requires all the variables considered to be strictly numbers. Which is not really the case with real life data. We get mix up of both numeric and string characteristics in our data. Then how do we use categorical variables in our model?

We need to figure out some way to convert them to numbers. Blindly giving them 1,2,3,4.... doesn't makes sense, it also makes your model simply bad. Because the mathematical framework is going to treat these numbers as usual. So 6 will be treated as 2 times 3, whereas say the variable in question was city names, and you assigned 3 to Agra and 6 to Kochi, it might not makes sense to say the **name** "Kochi" is 2 times "Agra".

We can comment that if I increase my numeric predictor say x_i by amount Δt then my target will change by some multiple of this say $\beta_i * \Delta t$. Now categorical variable don't "change" like that. At one time they will take some fixed category as a value at another some other category as a value. And measuring "change" in that context doesn't really make sense.

Now lets think how categorical variables really affect your target. Lets say your target is risk score for heart deceases and you have a model w.r.t. age As this:

$Risk = \beta_0 + \beta_1 * Age$

You have another categorical variable "history" which takes value "both" when both of your parents have history of having heart decease. "one" when one of the parents has history of heart decease, "none" otherwise. We can hypothetically say that, if history="both", Risk goes up by 0.10 or 10%, if history="one" then Risk goes up by 0.05 and no effect if history="none". So we can have three different models to incorporate this categorical variable.

when history="both" then $Risk = \beta_0 + \beta_1 * Age + 0.1$. if history="one" then $Risk = \beta_0 + \beta_1 * Age + 0.05$ and when history="none" then simply $Risk = \beta_0 + \beta_1 * Age$. We can combine this if we consider two **dummy** variables, history_both=1 when history="both" and 0 otherwise. history_one=1 when history="one" and 0 otherwise.

$Risk = \beta_0 + \beta_1 * Age + 0.1 * history\_both + 0.05 * history\_one$

The lesson here is that we can convert a categorical variable which takes **n** distinct values to a set of **n-1** dummy variables as mentioned above and use them just like numeric variables in the regression process. You must be wondering why **n-1** . Create on your own a hypothetical categorical variable which takes values "a" or "b" or "c" randomly. Create 10 observations for that. Now create three dummy variables cat_a, cat_b and cat_c. Observe that you can write:

$cat\_c = 1 - cat\_a - cat\_b$

you can randomly switch places of cat_a, cat_b and cat_c, and this equation will still hold. Basically if you know values of **n-1** dummy variables , you can perfectly know what would be the value of n^{th} dummy variable. To avoid this situation of perfect multicollineraity between predictors we need to make only **n-1** dummy variables for a categorical variable which takes **n** distinct values or has **n** distinct categories.

Then the question comes which one should I drop OR *not* create a dummy variable for.Well, it doesn't really matter. However, standard is to drop the category which is least frequent.

Model Validation

We have pretty much learned every thing which we needed to build a linear regression model. However it might not be enough still to achieve our goal , which was to come with an equation which enables us to forecast results for **yet unknown** or **future** values of predictors. Why? , because the parameter estimates that you got are for the data that you already have. How do you check if this model will perform well on the unseen data as well?

The answer is rather simple, randomly break your data in two samples and use one to build your model and check performance on the second one. We'll formally call them train and test datasets.

Although you'll come across few machine learning algorithms [Decision Trees], where they have cross validation inbuilt functions which utilize train data for tuning the model.

k fold cross validation : Underlying process is that , your train data is broken into k random parts. At a time a model is built on k-1 parts together and its performance [error] is checked on the left out part. This process is done k times , leaving one part [out of k parts] for measuring error. Final error is calculated as average of k iterations. This average error helps in tuning parameters of machine learning models. We'll implementation of cross validation when we reach to that part of the course.

For now we'll rely on breaking our data into two parts train and test.

Let me re-iterate why we need to do this. Reason is rather practical requirement than statistical. Ultimate goal of all this model building is to get an equation which can be used on unseen data to predict outcome of the business process.

Lets Summarise The Model Buidling Process

- Remove or impute missing values in the data
- Create dummy variables for the categorical variables
- Break your data in to three parts : train and test. start building model on train
  - In the first run of your model building process check VIF for all variables.
  - Drop variable with high VIF (>5). Drop them one by one only **not all at once**. drop the one with highest VIF, run your process again and then pick again the variable with highest VIF and drop, keep on doing this until VIF is (<5) for all variables.
  - Once VIF is under control for all the remaining variables , start dropping variable one by one based on p-values. If p-value is **greater** than 0.05, drop that variable
  - You have your train model , once you have all remaining variables with p-values less than 0.05.
- Test this model's performance by calculating RMSE (Root mean square error) on test dataset.
- You can use RMSE calculated on test data to compare multiple models.

Applications of Linear Regression :

Predictive Analytics: forecasting future opportunities and risks is the most prominent application of regression analysis in business. Demand analysis, for instance, predicts the number of items which a consumer will probably purchase. However, demand is not the only dependent variable when it comes to business. Regression analysis can go far beyond forecasting impact on direct revenue. For example, we can forecast the number of shoppers who will pass in front of a particular billboard and use that data to estimate the maximum to bid for an advertisement. Insurance companies heavily rely on regression analysis to estimate the credit standing of policyholders and a possible number of claims in a given time period.

New Insights: Over time businesses have gathered a large volume of unorganized data that has the potential to yield valuable insights. However, this data is useless without proper analysis. Regression analysis techniques can find a relationship between different variables by uncovering patterns that were previously unnoticed. For example, analysis of data from point of sales systems and purchase accounts may highlight market patterns like increase in demand on certain days of the week or at certain times of the year. You can maintain optimal stock and personnel before a spike in demand arises by acknowledging these insights.

Operation Efficiency: Regression models can also be used to optimize business processes. A factory manager, for example, can create a statistical model to understand the impact of oven temperature on the shelf life of the cookies baked in those ovens. In a call center, we can analyze the relationship between wait times of callers and number of complaints. Data-driven decision making eliminates guesswork, hypothesis and corporate politics from decision making. This improves the business performance by highlighting the areas that have the maximum impact on the operational efficiency and revenues.

Supporting Decisions: Businesses today are overloaded with data on finances, operations and customer purchases. Increasingly, executives are now leaning on data analytics to make informed business decisions thus eliminating the intuition and gut feel. Regression analysis can bring a scientific angle to the management of any businesses. By reducing the tremendous amount of raw data into actionable information, regression analysis leads the way to smarter and more accurate decisions. This does not mean that regression analysis is an end to managers creative thinking. This technique acts as a perfect tool to test a hypothesis before diving into execution.

Correcting Errors: Regression is not only great for lending empirical support to management decisions but also for identifying errors in judgment. For example, a retail store manager may believe that extending shopping hours will greatly increase sales. Regression analysis, however, may indicate that the increase in revenue might not be sufficient to support the rise in operating expenses due to longer working hours (such as additional employee labor charges). Hence, regression analysis can provide quantitative support for decisions and prevent mistakes due to manager's intuitions.

Case Study :

We'll also learn syntax for R [which is surprisingly simple] while we are carrying out model building process.

Loan Smart is a lending advisory firm. Based on their client's characteristic and needed loan amount they advise them on which Financial Institution to apply for loan at. So far their recommendations have been based hunches business experience. Now they are trying to leverage power of data that they have collected so far.

They want to check whether given their client's characteristics , they can predict how much interest rates they will be offered by various financial institution. They want to run with proof of concept for this idea. They have given us data collected for one such financial institution ABC Capitals Ltd.

What we need to do is to figure out whether using that data we can predict interest rate offered to client. We have developed the problem the way you'd encounter problems in projects. You are given training data and testing data, testing data doesnt have response values. We'll eventually want to make prediction on this data where the response is unknown. Lets start with importing data

```
ld_train=read.csv("~/Dropbox/0.0 Data/loan_data_train.csv",stringsAsFactors = F)

ld_test= read.csv("~/Dropbox/0.0 Data/loan_data_test.csv",stringsAsFactors = F)
library(dplyr)
glimpse(ld_train)
```

```
## Observations: 2,200
## Variables: 15
## $ ID                          <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested            <chr> "25000", "19750", "2100", "2800...
```

```
## $ Amount.Funded.By.Investors     <chr> "25000", "19750", "2100", "2800...
## $ Interest.Rate                  <chr> "18.49%", "17.27%", "14.33%", "...
## $ Loan.Length                    <chr> "60 months", "60 months", "36 m...
## $ Loan.Purpose                   <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio           <chr> "27.56%", "13.39%", "3.50%", "1...
## $ State                          <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                 <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                 <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ FICO.Range                     <chr> "720-724", "710-714", "690-694"...
## $ Open.CREDIT.Lines              <chr> "11", "14", "13", "12", "6", "2...
## $ Revolving.CREDIT.Balance       <chr> "15210", "19070", "893", "38194...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length              <chr> "5 years", "4 years", "< 1 year...
```

```
glimpse(ld_test)
```

```
## Observations: 300
## Variables: 14
## $ ID                             <int> 20093, 62445, 65248, 81822, 579...
## $ Amount.Requested               <int> 5000, 18000, 7200, 7200, 22000,...
## $ Amount.Funded.By.Investors     <chr> "5000", "18000", "7200", "7200"...
## $ Loan.Length                    <chr> "60 months", "60 months", "60 m...
## $ Loan.Purpose                   <chr> "moving", "debt_consolidation",...
## $ Debt.To.Income.Ratio           <chr> "12.59%", "4.93%", "25.16%", "1...
## $ State                          <chr> "NY", "CA", "LA", "NY", "MI", "...
## $ Home.Ownership                 <chr> "RENT", "RENT", "MORTGAGE", "MO...
## $ Monthly.Income                 <dbl> 4416.67, 5258.50, 3750.00, 3416...
## $ FICO.Range                     <chr> "690-694", "710-714", "750-754"...
## $ Open.CREDIT.Lines              <chr> "13", "6", "13", "14", "9", "."...
## $ Revolving.CREDIT.Balance       <int> 7686, 11596, 7283, 4838, 20181,...
## $ Inquiries.in.the.Last.6.Months <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0...
## $ Employment.Length              <chr> "< 1 year", "10+ years", "6 yea...
```

Variable names are self explanatory as to what they represent.In case you have any doubts , feel free to post on Q&A forum. Also Let me remind you at this point, a big chunk of predictive modeling is preparing your data, getting it ready for the modeling process.

We'll combine our two datasets so that we do not need to prepare data separately for them. And we'll also avoid problem of dealing with different columns in different datasets.

However before combining them, we'll need to add response column to test because number of columns need to be same for two datasets to stack vertically.

```
ld_test$Interest.Rate=NA

ld_train$data='train'
ld_test$data='test'

ld_all=rbind(ld_train,ld_test)
```

Now we'll start with data prep on ld_all.

When you look at the data set , you'll find out Interest.Rate , Debt.To.Income.Ratio have been imported as characters due to "%" sign . Open.Credit.Lines and some othe supposedly numeric variables have been converted to character becuase of some character value present possibly. Lets convert them to numbers first.

```
ld_all=ld_all %>%
  mutate(Interest.Rate=as.numeric(gsub("%","",Interest.Rate)) ,
```

```
         Debt.To.Income.Ratio=as.numeric(gsub("%","",Debt.To.Income.Ratio)) ,
         Open.CREDIT.Lines=as.numeric(Open.CREDIT.Lines) ,
         Amount.Requested=as.numeric(Amount.Requested) ,
         Amount.Funded.By.Investors=as.numeric(Amount.Funded.By.Investors),
         Revolving.CREDIT.Balance=as.numeric(Revolving.CREDIT.Balance)
         )
glimpse(ld_all)
```

```
## Observations: 2,500
## Variables: 16
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <dbl> 25000, 19750, 2100, 28000, 2425...
## $ Amount.Funded.By.Investors      <dbl> 25000.00, 19750.00, 2100.00, 28...
## $ Interest.Rate                   <dbl> 18.49, 17.27, 14.33, 16.29, 12....
## $ Loan.Length                     <chr> "60 months", "60 months", "36 m...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <dbl> 27.56, 13.39, 3.50, 19.62, 23.7...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                  <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ FICO.Range                      <chr> "720-724", "710-714", "690-694"...
## $ Open.CREDIT.Lines               <dbl> 11, 14, 13, 12, 6, 2, 5, 11, 24...
## $ Revolving.CREDIT.Balance        <dbl> 15210, 19070, 893, 38194, 31061...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length               <chr> "5 years", "4 years", "< 1 year...
## $ data                            <chr> "train", "train", "train", "tra...
```

The variable Amount_Funded_By_Investors happens to have high correlation with Interest.Rate [our target], but we should still drop this variables because it contains information which will not be available at the time when we need to use this model.

When someone comes with their loan application to Loan Smart , it does not contain information about amount funded by investors. This is a lesson for not falling in love with your data. Always think from a perspective of "At what point of the business process you are going to use this model" and then discard variables containing information which will not be available at that decision point.

```
ld_all = ld_all %>%
  select(-Amount.Funded.By.Investors)

# ld_all$Amount.Funded.By.Investors=NULL

# we could have used commented code instead for dropping. Infact i find commented method
# more convenient when dropping a single column. dplyr way is more conevnient when dropping
# multiple columns
```

lets see what we are going to do about categorical variables. Although fico_range is recorded as categorical variable in terms of ranges , we can convert it to a numeric variable by assigning value as average of the range. lets do that.

```
ld_all= ld_all %>%
  mutate(f1=as.numeric(substr(FICO.Range,1,3)),
         f2=as.numeric(substr(FICO.Range,5,7)),
         fico=0.5*(f1+f2)
         ) %>%
  select(-FICO.Range,-f1,-f2)
glimpse(ld_all)
```

```
## Observations: 2,500
## Variables: 15
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <dbl> 25000, 19750, 2100, 28000, 2425...
## $ Interest.Rate                   <dbl> 18.49, 17.27, 14.33, 16.29, 12....
## $ Loan.Length                     <chr> "60 months", "60 months", "36 m...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <dbl> 27.56, 13.39, 3.50, 19.62, 23.7...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                  <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ Open.CREDIT.Lines               <dbl> 11, 14, 13, 12, 6, 2, 5, 11, 24...
## $ Revolving.CREDIT.Balance        <dbl> 15210, 19070, 893, 38194, 31061...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length               <chr> "5 years", "4 years", "< 1 year...
## $ data                            <chr> "train", "train", "train", "tra...
## $ fico                            <dbl> 722, 712, 692, 712, 732, 787, 6...
```

In a similar fashion we can convert employment length to numbers as well.

```
ld_all=ld_all %>%
  mutate(el=ifelse(substr(Employment.Length,1,2)=="10",10,Employment.Length),
         el=ifelse(substr(Employment.Length,1,1)=="<",0,el),
         el=gsub("years","",el),
         el=gsub("year","",el),
         el=as.numeric(el)
         ) %>%
  select(-Employment.Length)
```

Lets take a pause at this point and think over the decision of how we processed employment length here. Converting this to numbers using the logic above was a subjective decision. We can't really say that this is what we do in every case. We might have as well went on treating it as categorical variable and created dummy variables for it. We might have used some other logic to convert it to numeric by assigning some other numeric values. Each of these decisions would have resulted in a different model. Lets say we build models m1, m2, m3 based on all these decision separately. You can select one of these models to go forward with , based on performance on the validation data. In this text however , we'll be focusing on just one iteration and in turn developing just one model. You can try out other models with different features/variables [arriving from different subjective decisions while preparing data] and see whether they perform better than the model developed in this iteration here. [Its not a challenge, your model can very well outperform the one developed here !]

Now lets resume and get to create dummy variables for our remaining categorical variables. We should ignore categories with very low count as well. You can choose not to ignore and make more dummy variables. That way you'll have one of your own iterations of this process and you can check whether your iteration performs better or not.

We'll be using the function CreateDummies that we wrote earlier

```
CreateDummies=function(data,var,freq_cutoff=0){
  t=table(data[,var])
  t=t[t>freq_cutoff]
  t=sort(t)
  categories=names(t)[-1]

  for( cat in categories){
    name=paste(var,cat,sep="_")
    name=gsub(" ","",name)
```

```
    name=gsub("-","_",name)
    name=gsub("\\?","Q",name)
    name=gsub("<","LT_",name)
    name=gsub("\\+","",name)

    data[,name]=as.numeric(data[,var]==cat)
  }

  data[,var]=NULL
  return(data)
}
```

Let me explain the function if you havent come across this before

`t=table(data[,var])` this bit creates a frequency table for the given categorical column. t here is now simply a table which contains names as categories of the categorical variable and their frequency in the data.

`t=t[t>freq_cutoff]` this line of code removes those categories from the table which have frequencies below the frequency cutoff. (this is a subjective choice)

't=sort(t)' this line simple sorts the remaining table in ascending order

`categories=names(t)[-1]` since we sorted the table in ascending manner in the previous line, first category here has least count. In this line of code we are taking out the category names except the first one (which has least count), thus making *n-1* dummies from the remaining categories.

`name=paste(var,cat,sep="_")` all the dummy vars that we intend to make, need to have some name. this line of code creates that name by concatenating variable name with category name with an `_`.

`name=gsub(" ","",name)` subsequent lines like these using gsub are essentially cleaning up the name thats all. Since we dont have any control over what the categories can be, we are removing special characters and spaces in the code in an automated fashion.

`data[,name]=as.numeric(data[,var]==cat)` once we have a cleaned up name, this line creates the dummy var for that particular category.

`data[,var]=NULL` once we are done creating dummies for the variable using for loop. Variable is removed from the data in this line.

Note: If you do choose to go ahead and not ignore the very less frequent category and make more dummy variables accrodingly , you'll notice that these dummy variables will end up having high VIF. Why?

Consider a hypothetical example where a categorical variable takes 5 distinct values a ,b ,c, d, e. Among these e occurs very few times. We do not make 5 dummy variable because of this relation ship:

$cat\_a + cat\_b + cat\_c + cat\_d + cat\_e = 1$

Which implies that if we make 5 dummy variables , any one taken at a time will be perfect linear combination of the rest which is undesirable as discussed previously.

So we will go ahead and take e as base category , this being least frequent and make 4 dummy variables. But e was a rare category. Meaning dummy variable for e , if made would have been equal to 1 rarely, which means that rest 4 dummies will still add up to exactly one for most part of the data , hence the high VIF values. We'll that in our results also.

Next we look at variable Loan.Purpose

```
table(ld_all$Loan.Purpose)
```

```
##
##               car       credit_card debt_consolidation
```

```
##                  50                 444                1307
##         educational    home_improvement               house
##                  15                 152                  20
##      major_purchase             medical              moving
##                 101                  30                  29
##               other    renewable_energy      small_business
##                 200                   4                  87
##            vacation             wedding
##                  21                  39
```

It has too many categories. There is no direct harm in considering to create dummy variables for n-1 for them. That can be one iteration to try. Here we are going to combine categories on the basis of average response , this will bring down number of categories and we can make dummies then.

```
round(tapply(ld_all$Interest.Rate,ld_all$Loan.Purpose,mean,na.rm=T))
```

```
##                 car         credit_card  debt_consolidation
##                  11                  13                  14
##         educational    home_improvement               house
##                  10                  12                  14
##      major_purchase             medical              moving
##                  11                  11                  14
##               other    renewable_energy      small_business
##                  13                   8                  13
##            vacation             wedding
##                  12                  12
```

we'll combine categories into new one which have similar reponse rate as per the table obtained abvoe and make dummies for them as well while we are at it.

```
ld_all=ld_all %>%
  mutate(lp_10=as.numeric(Loan.Purpose=='educational'),
         lp_11=as.numeric(Loan.Purpose %in% c("major_purchase","medical","car")),
         lp_12=as.numeric(Loan.Purpose %in% c("vacation","wedding","home_improvement")),
         lp_13=as.numeric(Loan.Purpose %in% c("other","small_business","credit_card")),
         lp_14=as.numeric(Loan.Purpose %in% c("debt_consolidation","house","moving"))) %>%
  select(-Loan.Purpose)
```

Next we will make dummy vars for remaining categorical variables.

```
for(col in c("Loan.Length","State","Home.Ownership")){
  ld_all=CreateDummies(ld_all,col,100)
}
```

Lets see if there are any missing values in the data.

```
lapply(ld_all,function(x) sum(is.na(x)))
```

```
## $ID
## [1] 1
##
## $Amount.Requested
## [1] 5
##
## $Interest.Rate
## [1] 300
##
## $Debt.To.Income.Ratio
```

```
## [1] 1
##
## $Monthly.Income
## [1] 3
##
## $Open.CREDIT.Lines
## [1] 9
##
## $Revolving.CREDIT.Balance
## [1] 5
##
## $Inquiries.in.the.Last.6.Months
## [1] 3
##
## $data
## [1] 0
##
## $fico
## [1] 0
##
## $el
## [1] 80
##
## $lp_10
## [1] 1
##
## $lp_11
## [1] 0
##
## $lp_12
## [1] 0
##
## $lp_13
## [1] 0
##
## $lp_14
## [1] 0
##
## $Loan.Length_36months
## [1] 1
##
## $State_FL
## [1] 1
##
## $State_TX
## [1] 1
##
## $State_NY
## [1] 1
##
## $State_CA
## [1] 1
##
## $Home.Ownership_RENT
```

```
## [1] 1
##
## $Home.Ownership_MORTGAGE
## [1] 1
```

It doesnt make sense to keep an observation with a missing ID, we'll filter that.

```
ld_all=ld_all[!(is.na(ld_all$ID)),]
```

Lets impute the missing values for remaining columns

```
for(col in names(ld_all)){

  if(sum(is.na(ld_all[,col]))>0 & !(col %in% c("ID","data","Interest.Rate"))){

    ld_all[is.na(ld_all[,col]),col]=mean(ld_all[ld_all$data=="train",col],na.rm=T)
  }

}
```

Notice that in missing values imputation also, only training data obs are used.

Now we are done with preparing data , lets separate the data.

```
ld_train=ld_all %>% filter(data=='train') %>% select(-data)
ld_test=ld_all %>% filter(data=='test') %>% select(-data,-Interest.Rate)
```

we can very well build the model on entire training data if we are not interested in looking at the performance of the model . However here we'd like to know how our model might tentatively perform on test/production data (it wont be an exact match ever, and in some cases realties might be very different). To do this, we'll break our data into two parts.

```
set.seed(2)
s=sample(1:nrow(ld_train),0.7*nrow(ld_train))
ld_train1=ld_train[s,]
ld_train2=ld_train[-s,]
```

Now we'll start with our modeling process. We'll be using function `lm` for building our linear regression model. First argument to this function is the modeling equation which we are trying to model. second argument is the dataset.

Lets say our data set has response y and predcitor variables as a ,b and c. then we'll write the modeling equation as this.

$$y \sim a + b + c$$

If the a, b, and c are the only predictor variables in the the data and you want to use all of them in your modeling equation, you can simply write:

$$y \sim .$$

If you want to use all predictor present in the data except , say b, then you can write :

$$y \sim . - b$$

Lets begin. Our response here is Interest.Rate and rest of the variables now in `ld_train` are our predictor variable. Before you begin make sure that none of the variables in the dataset which you are going to use for modeling are of character type.

```
glimpse(ld_train1)
```

```
## Observations: 1,539
## Variables: 22
## $ ID                              <int> 96693, 24038, 62644, 52129, 951...
## $ Amount.Requested                <dbl> 4750, 9250, 12000, 23000, 18000...
## $ Interest.Rate                   <dbl> 17.27, 18.79, 17.77, 7.62, 16.2...
## $ Debt.To.Income.Ratio            <dbl> 18.33, 19.27, 24.48, 6.51, 17.6...
## $ Monthly.Income                  <dbl> 3583.33, 4448.00, 3333.33, 2500...
## $ Open.CREDIT.Lines               <dbl> 11, 9, 12, 10, 11, 10, 17, 9, 1...
## $ Revolving.CREDIT.Balance        <dbl> 2964, 22704, 16984, 227, 9379, ...
## $ Inquiries.in.the.Last.6.Months <dbl> 2, 0, 0, 0, 1, 3, 0, 0, 0, 2, 0...
## $ fico                            <dbl> 757, 662, 667, 812, 672, 682, 7...
## $ el                              <dbl> 5, 10, 10, 10, 5, 2, 1, 3, 1, 9...
## $ lp_10                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ lp_11                           <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1...
## $ lp_12                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ lp_13                           <dbl> 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0...
## $ lp_14                           <dbl> 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0...
## $ Loan.Length_36months            <dbl> 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1...
## $ State_FL                        <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ State_TX                        <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0...
## $ State_NY                        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ State_CA                        <dbl> 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0...
## $ Home.Ownership_RENT             <dbl> 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0...
## $ Home.Ownership_MORTGAGE         <dbl> 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1...
```

You can see that all the variables in data are numeric , lets start now. It doesnt make sense to include variable `ID` in the modelling process. Its just a id number for the transactions which should be ignored.

```
fit=lm(Interest.Rate~. -ID,data=ld_train1)
```

Before we go start looking at model summary and p-value for the variables , we need to drop variables from the model which have high VIF. Remember that VIF is a mutual thing, so if you drop one variable, it might happen that VIF value for several other variables will go down. For this reason, always drop variables for high VIF , one by one, never in a chunk. For looking at VIF values for the variables , we'll use function `vif` from package `car`.

```
library(car)
vif(fit)
```

```
##                Amount.Requested           Debt.To.Income.Ratio
##                        1.587440                       1.376453
##                  Monthly.Income              Open.CREDIT.Lines
##                        1.398383                       1.291374
##       Revolving.CREDIT.Balance Inquiries.in.the.Last.6.Months
##                        1.310101                       1.049189
##                            fico                             el
##                        1.140807                       1.080287
##                           lp_10                          lp_11
##                        5.033328                      33.905532
##                           lp_12                          lp_13
##                       40.483526                     109.800863
##                           lp_14           Loan.Length_36months
##                      129.626575                       1.270940
##                        State_FL                       State_TX
```

```
##                        1.054117                        1.056787
##                        State_NY                        State_CA
##                        1.110128                        1.119225
##            Home.Ownership_RENT        Home.Ownership_MORTGAGE
##                        3.690806                        3.799728
```

It will be easier to identify high VIF vars if we sort the results and say look at top 3 only

```
sort(vif(fit),decreasing = T)[1:3]
```

```
##     lp_14     lp_13     lp_12
## 129.62658 109.80086  40.48353
```

We'll drop highest vif var and run the model again.

```
fit=lm(Interest.Rate~. -ID - lp_14,data=ld_train)
sort(vif(fit),decreasing = T)[1:3]
```

```
## Home.Ownership_MORTGAGE     Home.Ownership_RENT        Amount.Requested
##                3.735010                3.650675                1.587393
```

All VIF values are under control.Now we look at the model summary , specifically p-values asscociated with the variables. We can drop variables with high p-values [>0.05] one by one or we can use step function which drops vars based on AIC score one by one. Although the methodology is different but end result is generally similar due to both of them targetting vars which do not contribute towards explaning not very well.

```
fit=step(fit)
```

```
## Start:  AIC=3203.34
## Interest.Rate ~ (ID + Amount.Requested + Debt.To.Income.Ratio +
##     Monthly.Income + Open.CREDIT.Lines + Revolving.CREDIT.Balance +
##     Inquiries.in.the.Last.6.Months + fico + el + lp_10 + lp_11 +
##     lp_12 + lp_13 + lp_14 + Loan.Length_36months + State_FL +
##     State_TX + State_NY + State_CA + Home.Ownership_RENT + Home.Ownership_MORTGAGE) -
##     ID - lp_14
##
##                                  Df Sum of Sq     RSS    AIC
## - lp_12                           1       0.0  9267.7 3201.3
## - State_NY                        1       0.2  9267.9 3201.4
## - State_CA                        1       0.4  9268.1 3201.4
## - Debt.To.Income.Ratio            1       0.8  9268.5 3201.5
## - State_FL                        1       1.3  9269.0 3201.6
## - lp_11                           1       2.2  9269.9 3201.9
## - lp_10                           1       2.6  9270.3 3202.0
## - Home.Ownership_RENT             1       3.7  9271.5 3202.2
## - Revolving.CREDIT.Balance        1       5.6  9273.3 3202.7
## - el                              1       6.5  9274.2 3202.9
## - lp_13                           1       8.4  9276.1 3203.3
## <none>                                         9267.7 3203.3
## - Monthly.Income                  1      11.0  9278.7 3203.9
## - Home.Ownership_MORTGAGE         1      23.8  9291.5 3207.0
## - State_TX                        1      51.4  9319.1 3213.5
## - Open.CREDIT.Lines               1      55.4  9323.1 3214.4
## - Inquiries.in.the.Last.6.Months  1     383.3  9651.1 3290.5
## - Amount.Requested                1    2006.0 11273.7 3632.2
## - Loan.Length_36months            1    3129.4 12397.1 3841.1
## - fico                            1   17558.2 26826.0 5538.5
##
```

```
## Step:  AIC=3201.34
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_13 + Loan.Length_36months +
##     State_FL + State_TX + State_NY + State_CA + Home.Ownership_RENT +
##     Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - State_NY                        1       0.2  9267.9 3199.4
## - State_CA                        1       0.4  9268.1 3199.4
## - Debt.To.Income.Ratio            1       0.8  9268.5 3199.5
## - State_FL                        1       1.3  9269.0 3199.6
## - lp_11                           1       2.3  9270.0 3199.9
## - lp_10                           1       2.6  9270.4 3200.0
## - Home.Ownership_RENT             1       3.7  9271.5 3200.2
## - Revolving.CREDIT.Balance        1       5.6  9273.3 3200.7
## - el                              1       6.5  9274.2 3200.9
## <none>                                         9267.7 3201.3
## - lp_13                           1       8.9  9276.6 3201.4
## - Monthly.Income                  1      11.0  9278.8 3202.0
## - Home.Ownership_MORTGAGE         1      23.8  9291.5 3205.0
## - State_TX                        1      51.5  9319.2 3211.5
## - Open.CREDIT.Lines               1      55.4  9323.2 3212.5
## - Inquiries.in.the.Last.6.Months  1     383.5  9651.3 3288.5
## - Amount.Requested                1    2019.3 11287.0 3632.8
## - Loan.Length_36months            1    3132.3 12400.0 3839.6
## - fico                            1   17802.8 27070.5 5556.5
##
## Step:  AIC=3199.38
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_13 + Loan.Length_36months +
##     State_FL + State_TX + State_CA + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - State_CA                        1       0.5  9268.4 3197.5
## - Debt.To.Income.Ratio            1       0.8  9268.7 3197.6
## - State_FL                        1       1.2  9269.1 3197.7
## - lp_11                           1       2.3  9270.2 3197.9
## - lp_10                           1       2.6  9270.6 3198.0
## - Home.Ownership_RENT             1       3.8  9271.7 3198.3
## - Revolving.CREDIT.Balance        1       5.6  9273.5 3198.7
## - el                              1       6.6  9274.5 3198.9
## <none>                                         9267.9 3199.4
## - lp_13                           1       8.9  9276.8 3199.5
## - Monthly.Income                  1      10.9  9278.8 3200.0
## - Home.Ownership_MORTGAGE         1      24.7  9292.6 3203.2
## - State_TX                        1      51.5  9319.4 3209.6
## - Open.CREDIT.Lines               1      55.3  9323.2 3210.5
## - Inquiries.in.the.Last.6.Months  1     383.4  9651.3 3286.5
## - Amount.Requested                1    2020.1 11288.0 3631.0
## - Loan.Length_36months            1    3137.7 12405.6 3838.6
## - fico                            1   17802.6 27070.5 5554.5
##
```

```
## Step:  AIC=3197.51
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_13 + Loan.Length_36months +
##     State_FL + State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                 Df Sum of Sq     RSS    AIC
## - Debt.To.Income.Ratio           1       0.8  9269.2 3195.7
## - State_FL                       1       1.4  9269.8 3195.8
## - lp_11                          1       2.2  9270.7 3196.0
## - lp_10                          1       2.6  9271.0 3196.1
## - Home.Ownership_RENT            1       4.1  9272.5 3196.5
## - Revolving.CREDIT.Balance       1       5.7  9274.2 3196.9
## - el                             1       6.5  9275.0 3197.1
## <none>                                        9268.4 3197.5
## - lp_13                          1       9.0  9277.4 3197.6
## - Monthly.Income                 1      10.8  9279.3 3198.1
## - Home.Ownership_MORTGAGE        1      24.7  9293.1 3201.4
## - State_TX                       1      53.5  9322.0 3208.2
## - Open.CREDIT.Lines              1      55.4  9323.8 3208.6
## - Inquiries.in.the.Last.6.Months 1     385.2  9653.7 3285.1
## - Amount.Requested               1    2019.9 11288.3 3629.0
## - Loan.Length_36months           1    3147.0 12415.4 3838.3
## - fico                           1   17802.4 27070.9 5552.5
##
## Step:  AIC=3195.69
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_13 + Loan.Length_36months +
##     State_FL + State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                 Df Sum of Sq     RSS    AIC
## - State_FL                       1       1.4  9270.6 3194.0
## - lp_11                          1       2.4  9271.6 3194.3
## - lp_10                          1       2.8  9272.0 3194.4
## - Home.Ownership_RENT            1       4.0  9273.3 3194.7
## - el                             1       6.4  9275.6 3195.2
## - Revolving.CREDIT.Balance       1       6.7  9275.9 3195.3
## <none>                                        9269.2 3195.7
## - lp_13                          1       9.1  9278.3 3195.8
## - Monthly.Income                 1      10.1  9279.3 3196.1
## - Home.Ownership_MORTGAGE        1      24.5  9293.7 3199.5
## - State_TX                       1      52.9  9322.1 3206.2
## - Open.CREDIT.Lines              1      68.4  9337.6 3209.8
## - Inquiries.in.the.Last.6.Months 1     386.9  9656.1 3283.6
## - Amount.Requested               1    2023.6 11292.8 3627.9
## - Loan.Length_36months           1    3146.9 12416.1 3836.5
## - fico                           1   18301.5 27570.7 5590.7
##
## Step:  AIC=3194.03
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_13 + Loan.Length_36months +
##     State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
```

```
##
##                                 Df Sum of Sq     RSS    AIC
## - lp_11                          1       2.4  9273.1 3192.6
## - lp_10                          1       3.0  9273.6 3192.7
## - Home.Ownership_RENT            1       4.3  9274.9 3193.0
## - el                             1       6.3  9276.9 3193.5
## - Revolving.CREDIT.Balance       1       6.4  9277.0 3193.5
## <none>                                        9270.6 3194.0
## - lp_13                          1       9.2  9279.8 3194.2
## - Monthly.Income                 1      10.4  9281.0 3194.5
## - Home.Ownership_MORTGAGE        1      24.8  9295.4 3197.9
## - State_TX                       1      51.9  9322.5 3204.3
## - Open.CREDIT.Lines              1      68.6  9339.2 3208.2
## - Inquiries.in.the.Last.6.Months 1     386.2  9656.8 3281.8
## - Amount.Requested               1    2024.0 11294.7 3626.3
## - Loan.Length_36months           1    3145.7 12416.3 3834.5
## - fico                           1   18317.4 27588.1 5590.1
##
## Step:  AIC=3192.61
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_13 + Loan.Length_36months + State_TX +
##     Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                 Df Sum of Sq     RSS    AIC
## - lp_10                          1       2.7  9275.8 3191.2
## - Home.Ownership_RENT            1       4.4  9277.5 3191.7
## - el                             1       6.0  9279.0 3192.0
## - Revolving.CREDIT.Balance       1       6.6  9279.6 3192.2
## - lp_13                          1       7.8  9280.9 3192.5
## <none>                                        9273.1 3192.6
## - Monthly.Income                 1       9.5  9282.6 3192.9
## - Home.Ownership_MORTGAGE        1      25.5  9298.5 3196.6
## - State_TX                       1      52.6  9325.7 3203.1
## - Open.CREDIT.Lines              1      69.3  9342.3 3207.0
## - Inquiries.in.the.Last.6.Months 1     385.3  9658.3 3280.1
## - Amount.Requested               1    2073.0 11346.1 3634.3
## - Loan.Length_36months           1    3181.3 12454.4 3839.2
## - fico                           1   18654.6 27927.7 5615.0
##
## Step:  AIC=3191.25
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_13 + Loan.Length_36months + State_TX + Home.Ownership_RENT +
##     Home.Ownership_MORTGAGE
##
##                                 Df Sum of Sq     RSS    AIC
## - Home.Ownership_RENT            1       4.6  9280.4 3190.3
## - el                             1       5.8  9281.6 3190.6
## - Revolving.CREDIT.Balance       1       6.6  9282.4 3190.8
## - lp_13                          1       7.3  9283.1 3191.0
## <none>                                        9275.8 3191.2
## - Monthly.Income                 1       9.6  9285.4 3191.5
## - Home.Ownership_MORTGAGE        1      26.1  9301.8 3195.4
```

```
## - State_TX                          1      52.1  9327.9 3201.6
## - Open.CREDIT.Lines                 1      69.3  9345.1 3205.6
## - Inquiries.in.the.Last.6.Months    1     391.4  9667.2 3280.1
## - Amount.Requested                  1    2070.9 11346.7 3632.4
## - Loan.Length_36months              1    3179.0 12454.8 3837.3
## - fico                              1   18680.3 27956.1 5615.3
##
## Step:  AIC=3190.35
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_13 + Loan.Length_36months + State_TX + Home.Ownership_MORTGAGE
##
##                                   Df Sum of Sq     RSS    AIC
## - el                               1       6.2  9286.6 3189.8
## - Revolving.CREDIT.Balance         1       6.2  9286.7 3189.8
## - lp_13                            1       7.3  9287.8 3190.1
## <none>                                          9280.4 3190.3
## - Monthly.Income                   1       9.9  9290.3 3190.7
## - Home.Ownership_MORTGAGE          1      34.9  9315.3 3196.6
## - State_TX                         1      53.1  9333.5 3200.9
## - Open.CREDIT.Lines                1      68.4  9348.9 3204.5
## - Inquiries.in.the.Last.6.Months   1     392.0  9672.4 3279.3
## - Amount.Requested                 1    2069.3 11349.7 3631.0
## - Loan.Length_36months             1    3175.8 12456.2 3835.5
## - fico                             1   18727.5 28007.9 5617.3
##
## Step:  AIC=3189.81
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + lp_13 + Loan.Length_36months + State_TX + Home.Ownership_MORTGAGE
##
##                                   Df Sum of Sq     RSS    AIC
## - Revolving.CREDIT.Balance         1       5.7  9292.3 3189.2
## - lp_13                            1       7.5  9294.1 3189.6
## <none>                                          9286.6 3189.8
## - Monthly.Income                   1       9.9  9296.5 3190.2
## - Home.Ownership_MORTGAGE          1      30.3  9316.9 3195.0
## - State_TX                         1      51.2  9337.8 3199.9
## - Open.CREDIT.Lines                1      68.1  9354.7 3203.9
## - Inquiries.in.the.Last.6.Months   1     388.5  9675.1 3277.9
## - Amount.Requested                 1    2107.7 11394.3 3637.6
## - Loan.Length_36months             1    3176.8 12463.4 3834.8
## - fico                             1   18751.9 28038.5 5617.7
##
## Step:  AIC=3189.16
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + lp_13 + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
##
##                                   Df Sum of Sq     RSS    AIC
## - lp_13                            1       6.4  9298.7 3188.7
## <none>                                          9292.3 3189.2
## - Monthly.Income                   1      14.9  9307.2 3190.7
## - Home.Ownership_MORTGAGE          1      33.2  9325.5 3195.0
```

```
## - State_TX                          1      52.5  9344.8 3199.6
## - Open.CREDIT.Lines                 1      80.1  9372.3 3206.0
## - Inquiries.in.the.Last.6.Months    1     389.8  9682.1 3277.5
## - Amount.Requested                  1    2119.6 11411.9 3639.0
## - Loan.Length_36months              1    3191.2 12483.5 3836.3
## - fico                              1   18755.2 28047.4 5616.4
##
## Step:  AIC=3188.68
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
##
##                                 Df Sum of Sq     RSS    AIC
## <none>                                        9298.7 3188.7
## - Monthly.Income                 1      15.2  9313.9 3190.3
## - Home.Ownership_MORTGAGE        1      34.2  9333.0 3194.8
## - State_TX                       1      53.1  9351.8 3199.2
## - Open.CREDIT.Lines              1      78.9  9377.6 3205.3
## - Inquiries.in.the.Last.6.Months 1     388.4  9687.1 3276.7
## - Amount.Requested               1    2113.3 11412.1 3637.0
## - Loan.Length_36months           1    3184.8 12483.6 3834.4
## - fico                           1   18803.2 28101.9 5618.7
```

lets check model summary to see what all variable remain in the model.

```
summary(fit)
```

```
##
## Call:
## lm(formula = Interest.Rate ~ Amount.Requested + Monthly.Income +
##     Open.CREDIT.Lines + Inquiries.in.the.Last.6.Months + fico +
##     Loan.Length_36months + State_TX + Home.Ownership_MORTGAGE,
##     data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5280 -1.3681 -0.2007  1.2432 10.0654
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    7.466e+01  9.250e-01  80.718  < 2e-16 ***
## Amount.Requested               1.509e-04  6.764e-06  22.310  < 2e-16 ***
## Monthly.Income                -2.275e-05  1.203e-05  -1.892 0.058675 .
## Open.CREDIT.Lines             -4.414e-02  1.024e-02  -4.311  1.7e-05 ***
## Inquiries.in.the.Last.6.Months 3.504e-01  3.664e-02   9.564  < 2e-16 ***
## fico                          -8.562e-02  1.287e-03 -66.547  < 2e-16 ***
## Loan.Length_36months          -3.233e+00  1.180e-01 -27.388  < 2e-16 ***
## State_TX                       6.253e-01  1.769e-01   3.536 0.000415 ***
## Home.Ownership_MORTGAGE       -2.623e-01  9.238e-02  -2.839 0.004566 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.061 on 2190 degrees of freedom
## Multiple R-squared:  0.7571, Adjusted R-squared:  0.7562
## F-statistic: 853.4 on 8 and 2190 DF,  p-value: < 2.2e-16
```

It turns out there is still one var for which p-value is higher than .05 . we can manually iclude all other vars of the model or extract the exact formula using `formula` function.

```
formula(fit)
```

```
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
```

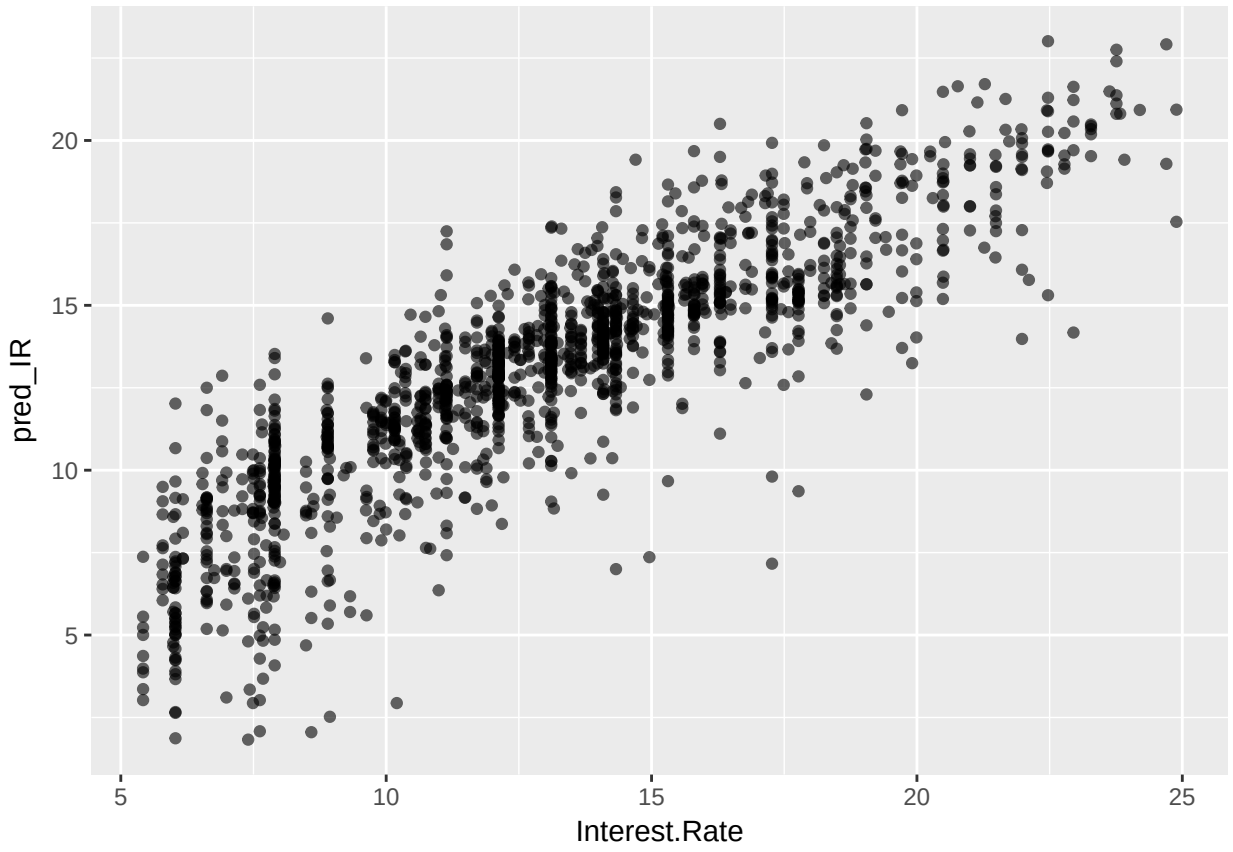
We can drop variable Monthly.Income from this.

```
fit=lm(Interest.Rate ~ Amount.Requested  + Open.CREDIT.Lines +
    Inquiries.in.the.Last.6.Months + fico + Loan.Length_36months +
    State_TX + Home.Ownership_MORTGAGE,data=ld_train1)
summary(fit)
```

```
##
## Call:
## lm(formula = Interest.Rate ~ Amount.Requested + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE, data = ld_train1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1051 -1.3524 -0.2032  1.2105 10.1054
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    7.463e+01  1.111e+00  67.199  < 2e-16 ***
## Amount.Requested               1.398e-04  7.706e-06  18.145  < 2e-16 ***
## Open.CREDIT.Lines             -2.996e-02  1.210e-02  -2.476 0.013398 *
## Inquiries.in.the.Last.6.Months  3.788e-01  4.522e-02   8.378  < 2e-16 ***
## fico                           -8.553e-02  1.536e-03 -55.670  < 2e-16 ***
## Loan.Length_36months           -3.426e+00  1.442e-01 -23.766  < 2e-16 ***
## State_TX                        7.041e-01  2.137e-01   3.295 0.001006 **
## Home.Ownership_MORTGAGE        -3.838e-01  1.104e-01  -3.477 0.000521 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.078 on 1531 degrees of freedom
## Multiple R-squared:  0.7575, Adjusted R-squared:  0.7564
## F-statistic: 683.3 on 7 and 1531 DF,  p-value: < 2.2e-16
```

This is our final model with all variables being significant [p-value < 0.05]. How good is this model. Adj R-squared value for the model is 0.7564 which means that this models explains 75.64% variation in our response. You can quickly check this visually too by plotting predicted values and the original response.

```
library(ggplot2)
ld_train1 %>%
  mutate(pred_IR=predict(fit,newdata=ld_train1)) %>%
  ggplot(aes(x=Interest.Rate,y=pred_IR))+geom_point(alpha=0.6)
```



You can see that there is good overlap.

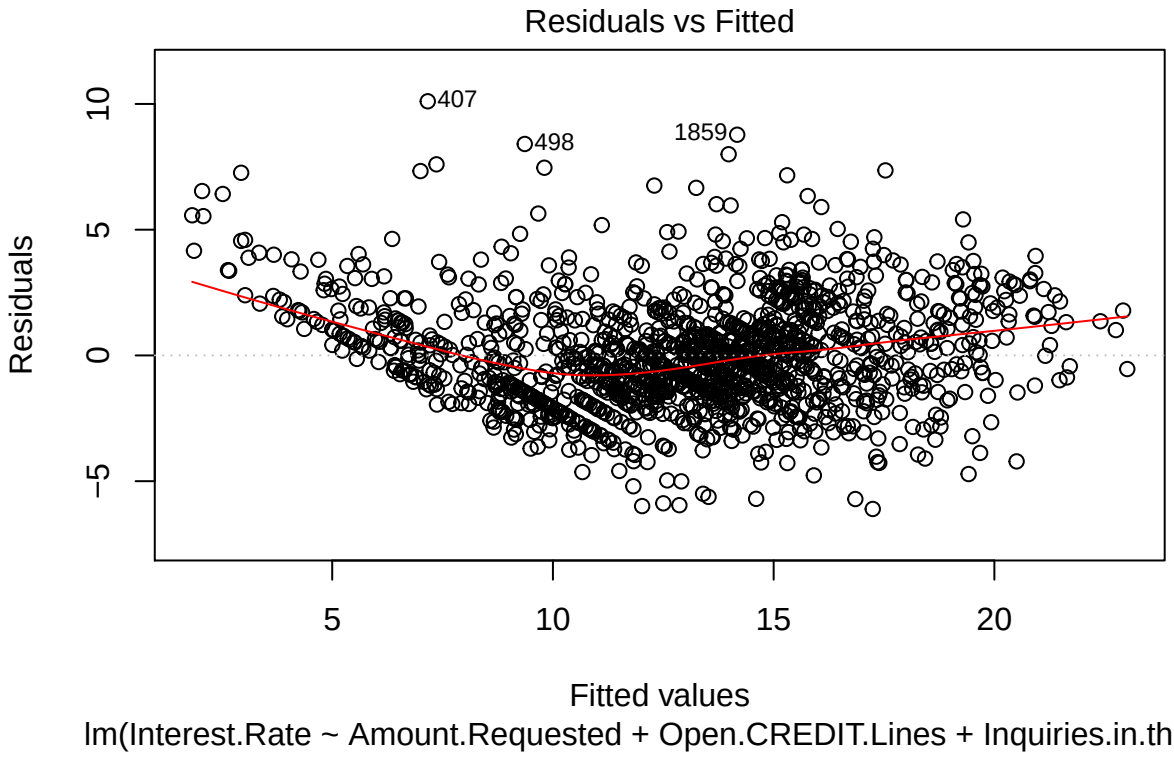
Our model equation from the model is :[taking down values from estimates from `summary(fit)`]. We are going to use collapse property of paste function. The result will need little modification but it saves us from noting down each value and writing prediction equation manually.

```
model_string=paste(fit$coefficients,names(fit$coefficients),sep="*",collapse = " + ")
model_eq=strwrap(sub("\\*\\(Intercept\\)","",gsub("+ -","- ",model_string,fixed=TRUE)))
model_eq
```

```
## [1] "74.6315269130068 + 0.000139829094726773*Amount.Requested -"
## [2] "0.0299642785221752*Open.CREDIT.Lines +"
## [3] "0.378824273745444*Inquiries.in.the.Last.6.Months -"
## [4] "0.0855341745510592*fico - 3.42594570009614*Loan.Length_36months +"
## [5] "0.704051024307523*State_TX -"
## [6] "0.383847331337859*Home.Ownership_MORTGAGE"
```

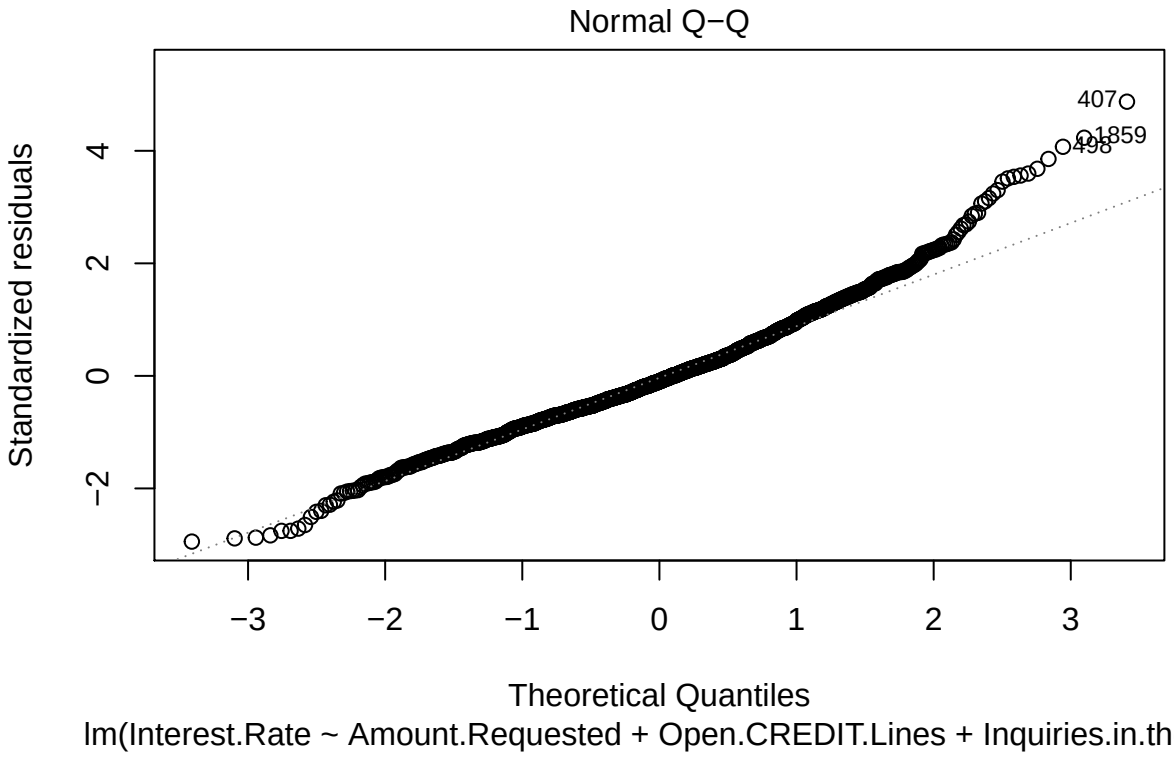
Next we look at different diagnostic plots to see if any of our linear regression assumptions are being violated and some other information regarding outliers.

```
plot(fit,which=1)
```



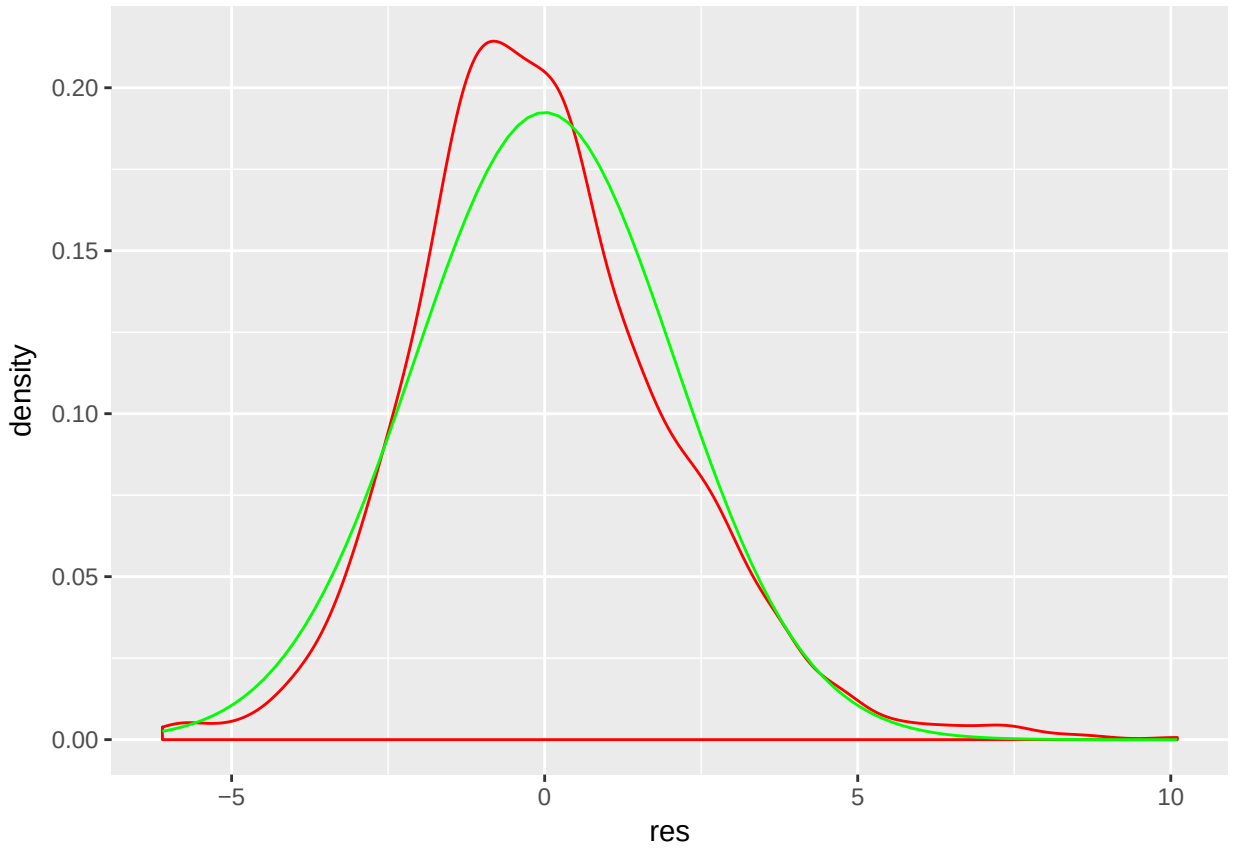
Ideally the red line the plot should be horizontal or close to horizontal . If it shows a significant pattern , it means we might have missed non-linear components in the model. If that is the case you might want to look at pair wise relation ship between response and each predictor and figure out if any of the predictors need to be transformed to make the relationship linear between response and that predictor. [Although a better alternative will be to go ahead and to use an algorithm which can extract non linear relationship. e.g. RandomForest , BoostinMachines etc]

```
plot(fit,which=2)
```



This tells you whether you residuals are normal or not. You can visualise/see this by plotting density plots as well.

```
df=data.frame(res=fit$residual)
ggplot(df,aes(x=res))+geom_density(color="red")+
  stat_function(fun=dnorm ,args = list(mean=mean(df$res),sd=sd(df$res)),color="green")
```

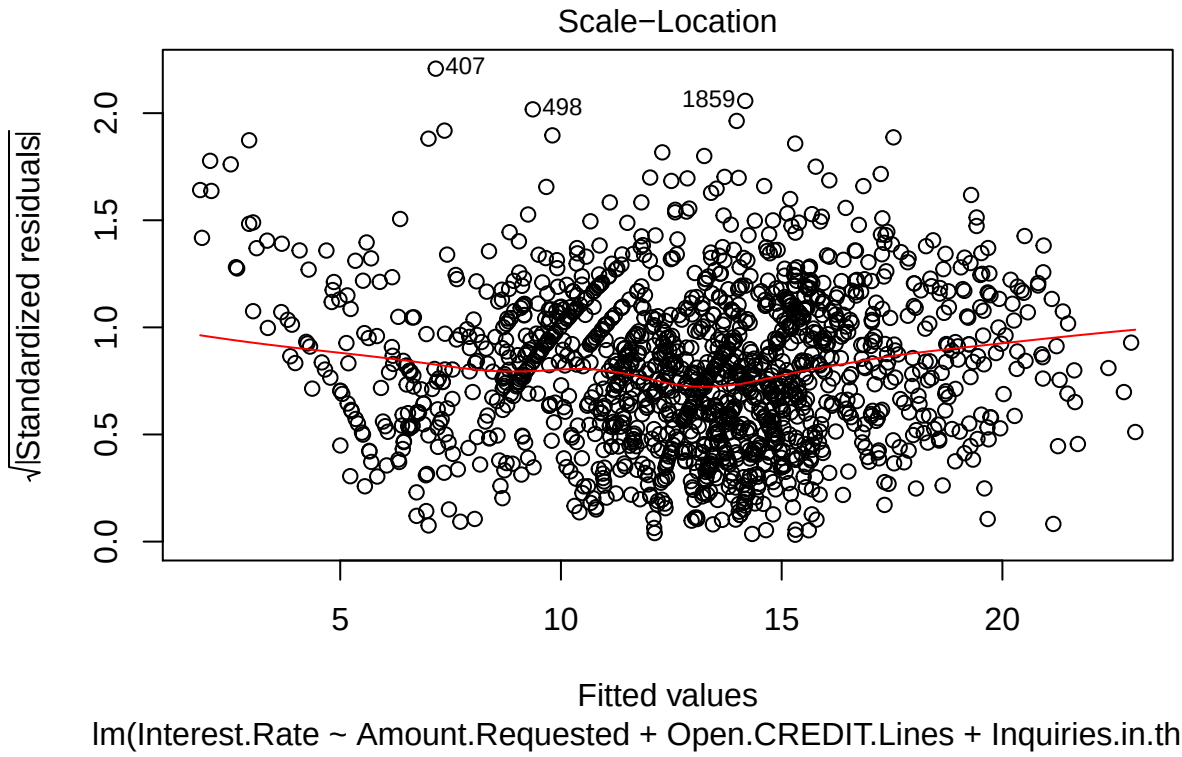



```
shapiro.test(fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.97921, p-value = 3.951e-14
```

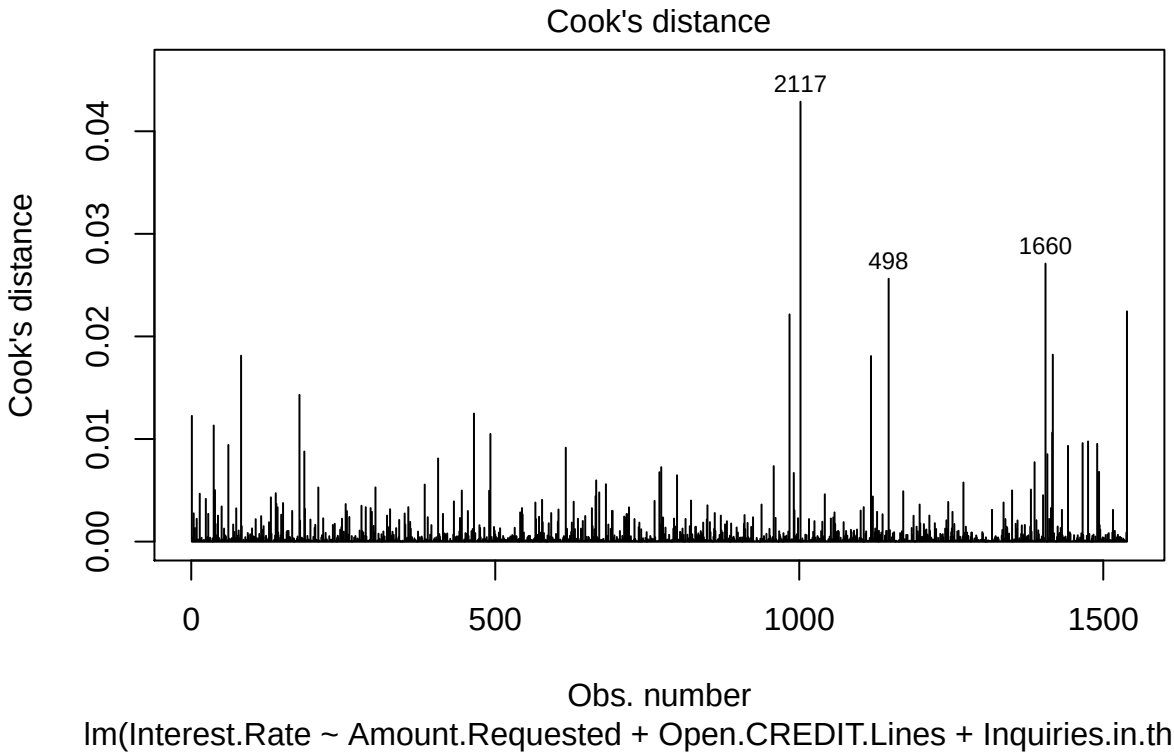
This tells you that there is significant deviation from normality in your residual which is confirmed by shapiro wilk test as well. This essentially means that you should not use p-values to drop vars as they rely on the assumption of normality. [We should have only used AIC score to drop vars]

```
plot(fit,which=3)
```



This plot tells us whether our residuals are homoscedastic or not. If this plot has a pattern then that mean residuals are heteroscedastic. If that is the case which is not in our case , you can again log transform response and rebuild your model. Why log transform of response in these deviations of residuals? Becuase it brings down scale of response and in turn for residuals. Effect of scale coming down is differences being less prominent.

```
plot(fit,which=4)
```



This gives us cook's distance for all the observations. If cook's distance for any observation is greater than 1 , you should consider removing that observation from your modeling process and rebuild the model. In our case; cook's distance is less than 1 for all the observations.[dont worry about high cook's distance value for any observation if its less than 1]

Cook's distance basically tells how much leverage [effect] one observation has on your model equation. High value of cook's distance [>1] hints at that observation being very different from rest of the data or in other words, that observation being an outlier.

Done with the diagnostic plots.But we havent yet done the crucial part of testing here. Lets look at rmse value for validation data

```
rmse= mean((ld_train2$Interest.Rate-predict(fit,newdata=ld_train2))**2) %>%
sqrt()
rmse
```

```
## [1] 2.034159
```

We see that model performance [RMSE] on validation data comes out to be 2.03. We can check performance of other models with this and then choose the best one having least RMSE.

RMSE can be used for benchmarking of your model. If you build another model , say using different features , or techniques [such as DTrees, RandomForest etc], you can compare RMSEs for them with the one that you got here and select the model for production which has best RMSE.

Note: Remeber that when you are comparing RMSEs for models, train and val data for them should be having same observations. As in , same set of datasets should be used across algorithms.

RMSE as a stand alone value doesnt hold much meaning and there is no set standard, against which you can compare and conclude that its a good/bad RMSE value that you got.

Overfit & Underfit : Performance on train and validation should not be vastly different. [Not true for high bias algortihm such as Decision Trees]. Models which perform very well on train data and not so well on validation data are said to be overfitting.

On the other hand models which perform much better on validation in comparison to train are called to be underfit. This is not a bad thing as long as you select models based on performance on validation data.

How ever overfitting and underfitting of the model gives you hint on how you can improve your model.

Final model

Now that we know about the tentative performance of the model , we'll go ahead and build the model on the entire training data and use that model to predict result on the test/production data.

```
fit.final=fit=lm(Interest.Rate ~ .-ID,
                 data=ld_train)
fit.final=step(fit.final)
```

```
## Start:  AIC=3204.82
## Interest.Rate ~ (ID + Amount.Requested + Debt.To.Income.Ratio +
##     Monthly.Income + Open.CREDIT.Lines + Revolving.CREDIT.Balance +
##     Inquiries.in.the.Last.6.Months + fico + el + lp_10 + lp_11 +
##     lp_12 + lp_13 + lp_14 + Loan.Length_36months + State_FL +
##     State_TX + State_NY + State_CA + Home.Ownership_RENT + Home.Ownership_MORTGAGE) -
##     ID
##
##                                  Df Sum of Sq     RSS    AIC
## - State_NY                        1       0.2  9265.7 3202.9
## - lp_10                           1       0.3  9265.8 3202.9
## - State_CA                        1       0.3  9265.9 3202.9
## - Debt.To.Income.Ratio            1       0.8  9266.3 3203.0
## - State_FL                        1       1.3  9266.9 3203.1
## - lp_13                           1       1.4  9266.9 3203.2
## - lp_11                           1       1.4  9267.0 3203.2
## - lp_12                           1       2.2  9267.7 3203.3
## - lp_14                           1       2.2  9267.7 3203.3
## - Home.Ownership_RENT             1       3.8  9269.3 3203.7
## - Revolving.CREDIT.Balance        1       5.6  9271.1 3204.2
## - el                              1       6.6  9272.1 3204.4
## <none>                                         9265.5 3204.8
## - Monthly.Income                  1      10.9  9276.4 3205.4
## - Home.Ownership_MORTGAGE         1      24.0  9289.5 3208.5
## - State_TX                        1      51.7  9317.3 3215.1
## - Open.CREDIT.Lines               1      54.5  9320.1 3215.7
## - Inquiries.in.the.Last.6.Months  1     383.3  9648.8 3292.0
## - Amount.Requested                1    2008.1 11273.7 3634.2
## - Loan.Length_36months            1    3130.7 12396.3 3842.9
## - fico                            1   17547.1 26812.7 5539.4
##
## Step:  AIC=3202.86
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_10 + lp_11 + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_FL + State_TX + State_CA + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - lp_10                           1       0.3  9266.0 3200.9
## - State_CA                        1       0.5  9266.2 3201.0
```

```
## - Debt.To.Income.Ratio            1       0.8  9266.6 3201.1
## - State_FL                        1       1.2  9267.0 3201.2
## - lp_13                           1       1.4  9267.1 3201.2
## - lp_11                           1       1.4  9267.2 3201.2
## - lp_12                           1       2.1  9267.9 3201.4
## - lp_14                           1       2.2  9267.9 3201.4
## - Home.Ownership_RENT             1       3.8  9269.6 3201.8
## - Revolving.CREDIT.Balance        1       5.6  9271.3 3202.2
## - el                              1       6.7  9272.4 3202.4
## <none>                                         9265.7 3202.9
## - Monthly.Income                  1      10.7  9276.5 3203.4
## - Home.Ownership_MORTGAGE         1      24.9  9290.6 3206.8
## - State_TX                        1      51.7  9317.4 3213.1
## - Open.CREDIT.Lines               1      54.3  9320.1 3213.7
## - Inquiries.in.the.Last.6.Months  1     383.1  9648.8 3290.0
## - Amount.Requested                1    2008.6 11274.3 3632.3
## - Loan.Length_36months            1    3135.8 12401.5 3841.9
## - fico                            1   17547.1 26812.8 5537.4
##
## Step:  AIC=3200.93
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_11 + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_FL + State_TX + State_CA + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - State_CA                        1       0.5  9266.5 3199.0
## - Debt.To.Income.Ratio            1       0.8  9266.8 3199.1
## - State_FL                        1       1.2  9267.2 3199.2
## - lp_11                           1       2.3  9268.3 3199.5
## - lp_13                           1       2.3  9268.3 3199.5
## - Home.Ownership_RENT             1       3.8  9269.8 3199.8
## - lp_12                           1       4.2  9270.2 3199.9
## - lp_14                           1       4.6  9270.5 3200.0
## - Revolving.CREDIT.Balance        1       5.6  9271.6 3200.3
## - el                              1       6.7  9272.7 3200.5
## <none>                                         9266.0 3200.9
## - Monthly.Income                  1      10.8  9276.8 3201.5
## - Home.Ownership_MORTGAGE         1      24.7  9290.7 3204.8
## - State_TX                        1      51.7  9317.7 3211.2
## - Open.CREDIT.Lines               1      54.7  9320.7 3211.9
## - Inquiries.in.the.Last.6.Months  1     382.8  9648.8 3288.0
## - Amount.Requested                1    2009.1 11275.1 3630.5
## - Loan.Length_36months            1    3135.6 12401.6 3839.9
## - fico                            1   17547.1 26813.1 5535.4
##
## Step:  AIC=3199.05
## Interest.Rate ~ Amount.Requested + Debt.To.Income.Ratio + Monthly.Income +
##     Open.CREDIT.Lines + Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_11 + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_FL + State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - Debt.To.Income.Ratio            1       0.8  9267.2 3197.2
```

```
## - State_FL                         1       1.4  9267.9 3197.4
## - lp_11                            1       2.3  9268.8 3197.6
## - lp_13                            1       2.3  9268.8 3197.6
## - Home.Ownership_RENT              1       4.1  9270.5 3198.0
## - lp_12                            1       4.2  9270.7 3198.0
## - lp_14                            1       4.5  9271.0 3198.1
## - Revolving.CREDIT.Balance         1       5.8  9272.2 3198.4
## - el                               1       6.6  9273.1 3198.6
## <none>                                          9266.5 3199.0
## - Monthly.Income                   1      10.7  9277.2 3199.6
## - Home.Ownership_MORTGAGE          1      24.8  9291.3 3202.9
## - State_TX                         1      53.7  9320.2 3209.7
## - Open.CREDIT.Lines                1      54.9  9321.3 3210.0
## - Inquiries.in.the.Last.6.Months   1     384.6  9651.1 3286.5
## - Amount.Requested                 1    2008.7 11275.2 3628.5
## - Loan.Length_36months             1    3144.5 12411.0 3839.5
## - fico                             1   17546.6 26813.1 5533.5
##
## Step:  AIC=3197.23
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_11 + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_FL + State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                   Df Sum of Sq     RSS    AIC
## - State_FL                         1       1.4  9268.7 3195.6
## - lp_11                            1       2.3  9269.6 3195.8
## - lp_13                            1       2.4  9269.7 3195.8
## - Home.Ownership_RENT              1       4.0  9271.3 3196.2
## - lp_12                            1       4.3  9271.5 3196.2
## - lp_14                            1       4.7  9272.0 3196.4
## - el                               1       6.5  9273.7 3196.8
## - Revolving.CREDIT.Balance         1       6.7  9273.9 3196.8
## <none>                                          9267.2 3197.2
## - Monthly.Income                   1      10.0  9277.2 3197.6
## - Home.Ownership_MORTGAGE          1      24.6  9291.8 3201.0
## - State_TX                         1      53.1  9320.3 3207.8
## - Open.CREDIT.Lines                1      67.4  9334.6 3211.2
## - Inquiries.in.the.Last.6.Months   1     386.2  9653.4 3285.0
## - Amount.Requested                 1    2011.4 11278.7 3627.2
## - Loan.Length_36months             1    3144.2 12411.5 3837.6
## - fico                             1   17978.9 27246.2 5566.7
##
## Step:  AIC=3195.56
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_11 + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                   Df Sum of Sq     RSS    AIC
## - lp_11                            1       2.4  9271.1 3194.1
## - lp_13                            1       2.5  9271.2 3194.2
## - Home.Ownership_RENT              1       4.3  9272.9 3194.6
## - lp_12                            1       4.5  9273.1 3194.6
```

```
## - lp_14                           1       4.9  9273.6 3194.7
## - el                              1       6.3  9275.0 3195.1
## - Revolving.CREDIT.Balance        1       6.4  9275.0 3195.1
## <none>                                         9268.7 3195.6
## - Monthly.Income                  1      10.3  9279.0 3196.0
## - Home.Ownership_MORTGAGE         1      24.9  9293.6 3199.5
## - State_TX                        1      52.1  9320.7 3205.9
## - Open.CREDIT.Lines               1      67.6  9336.3 3209.5
## - Inquiries.in.the.Last.6.Months  1     385.5  9654.2 3283.2
## - Amount.Requested                1    2011.5 11280.2 3625.5
## - Loan.Length_36months            1    3143.0 12411.6 3835.7
## - fico                            1   17990.7 27259.4 5565.8
##
## Step:  AIC=3194.14
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_12 + lp_13 + lp_14 + Loan.Length_36months +
##     State_TX + Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - lp_13                           1       0.1  9271.3 3192.2
## - lp_12                           1       2.7  9273.8 3192.8
## - Home.Ownership_RENT             1       4.4  9275.5 3193.2
## - lp_14                           1       4.6  9275.7 3193.2
## - el                              1       6.3  9277.4 3193.6
## - Revolving.CREDIT.Balance        1       6.4  9277.5 3193.6
## <none>                                         9271.1 3194.1
## - Monthly.Income                  1      10.7  9281.8 3194.7
## - Home.Ownership_MORTGAGE         1      25.1  9296.2 3198.1
## - State_TX                        1      51.4  9322.5 3204.3
## - Open.CREDIT.Lines               1      67.9  9339.0 3208.2
## - Inquiries.in.the.Last.6.Months  1     390.9  9662.0 3283.0
## - Amount.Requested                1    2012.1 11283.2 3624.1
## - Loan.Length_36months            1    3141.4 12412.6 3833.8
## - fico                            1   17988.4 27259.5 5563.8
##
## Step:  AIC=3192.18
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_12 + lp_14 + Loan.Length_36months + State_TX +
##     Home.Ownership_RENT + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - lp_12                           1       3.3  9274.6 3191.0
## - Home.Ownership_RENT             1       4.4  9275.7 3191.2
## - el                              1       6.2  9277.4 3191.6
## - Revolving.CREDIT.Balance        1       6.5  9277.8 3191.7
## <none>                                         9271.3 3192.2
## - Monthly.Income                  1      10.6  9281.8 3192.7
## - lp_14                           1      11.2  9282.5 3192.8
## - Home.Ownership_MORTGAGE         1      25.2  9296.4 3196.1
## - State_TX                        1      51.4  9322.7 3202.3
## - Open.CREDIT.Lines               1      68.2  9339.4 3206.3
## - Inquiries.in.the.Last.6.Months  1     391.1  9662.3 3281.0
```

```
## - Amount.Requested                1    2047.4 11318.6 3628.9
## - Loan.Length_36months            1    3165.5 12436.7 3836.1
## - fico                            1   18411.2 27682.4 5595.6
##
## Step:  AIC=3190.96
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_14 + Loan.Length_36months + State_TX + Home.Ownership_RENT +
##     Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - Home.Ownership_RENT             1       4.1  9278.7 3189.9
## - Revolving.CREDIT.Balance        1       6.1  9280.7 3190.4
## - el                              1       6.3  9280.9 3190.5
## <none>                                         9274.6 3191.0
## - lp_14                           1       8.5  9283.1 3191.0
## - Monthly.Income                  1      11.1  9285.7 3191.6
## - Home.Ownership_MORTGAGE         1      25.6  9300.2 3195.0
## - State_TX                        1      51.2  9325.8 3201.1
## - Open.CREDIT.Lines               1      67.2  9341.8 3204.8
## - Inquiries.in.the.Last.6.Months  1     389.4  9664.0 3279.4
## - Amount.Requested                1    2045.2 11319.8 3627.2
## - Loan.Length_36months            1    3162.5 12437.1 3834.2
## - fico                            1   18550.7 27825.3 5604.9
##
## Step:  AIC=3189.93
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Revolving.CREDIT.Balance + Inquiries.in.the.Last.6.Months +
##     fico + el + lp_14 + Loan.Length_36months + State_TX + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - Revolving.CREDIT.Balance        1       5.8  9284.4 3189.3
## - el                              1       6.7  9285.4 3189.5
## <none>                                         9278.7 3189.9
## - lp_14                           1       9.1  9287.8 3190.1
## - Monthly.Income                  1      11.5  9290.1 3190.6
## - Home.Ownership_MORTGAGE         1      36.3  9315.0 3196.5
## - State_TX                        1      52.1  9330.8 3200.2
## - Open.CREDIT.Lines               1      66.3  9345.0 3203.6
## - Inquiries.in.the.Last.6.Months  1     389.9  9668.5 3278.4
## - Amount.Requested                1    2045.1 11323.8 3625.9
## - Loan.Length_36months            1    3159.5 12438.1 3832.3
## - fico                            1   18588.0 27866.7 5606.2
##
## Step:  AIC=3189.3
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + el + lp_14 + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - el                              1       6.2  9290.6 3188.8
## <none>                                         9284.4 3189.3
## - lp_14                           1       8.5  9292.9 3189.3
## - Monthly.Income                  1      16.7  9301.1 3191.3
```

```
## - Home.Ownership_MORTGAGE         1      38.9  9323.4 3196.5
## - State_TX                        1      53.4  9337.8 3199.9
## - Open.CREDIT.Lines               1      78.4  9362.8 3205.8
## - Inquiries.in.the.Last.6.Months  1     391.2  9675.6 3278.1
## - Amount.Requested                1    2054.7 11339.1 3626.9
## - Loan.Length_36months            1    3176.2 12460.7 3834.3
## - fico                            1   18597.8 27882.2 5605.4
##
## Step:  AIC=3188.76
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + lp_14 + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## - lp_14                           1       8.1  9298.7 3188.7
## <none>                                        9290.6 3188.8
## - Monthly.Income                  1      16.4  9307.0 3190.6
## - Home.Ownership_MORTGAGE         1      34.1  9324.7 3194.8
## - State_TX                        1      51.5  9342.1 3198.9
## - Open.CREDIT.Lines               1      77.6  9368.2 3205.0
## - Inquiries.in.the.Last.6.Months  1     387.7  9678.3 3276.7
## - Amount.Requested                1    2093.6 11384.2 3633.7
## - Loan.Length_36months            1    3176.7 12467.3 3833.5
## - fico                            1   18621.3 27911.9 5605.8
##
## Step:  AIC=3188.68
## Interest.Rate ~ Amount.Requested + Monthly.Income + Open.CREDIT.Lines +
##     Inquiries.in.the.Last.6.Months + fico + Loan.Length_36months +
##     State_TX + Home.Ownership_MORTGAGE
##
##                                  Df Sum of Sq     RSS    AIC
## <none>                                        9298.7 3188.7
## - Monthly.Income                  1      15.2  9313.9 3190.3
## - Home.Ownership_MORTGAGE         1      34.2  9333.0 3194.8
## - State_TX                        1      53.1  9351.8 3199.2
## - Open.CREDIT.Lines               1      78.9  9377.6 3205.3
## - Inquiries.in.the.Last.6.Months  1     388.4  9687.1 3276.7
## - Amount.Requested                1    2113.3 11412.1 3637.0
## - Loan.Length_36months            1    3184.8 12483.6 3834.4
## - fico                            1   18803.2 28101.9 5618.7
```

```
summary(fit.final)
```

```
##
## Call:
## lm(formula = Interest.Rate ~ Amount.Requested + Monthly.Income +
##     Open.CREDIT.Lines + Inquiries.in.the.Last.6.Months + fico +
##     Loan.Length_36months + State_TX + Home.Ownership_MORTGAGE,
##     data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5280 -1.3681 -0.2007  1.2432 10.0654
##
## Coefficients:
```

```
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.466e+01  9.250e-01  80.718  < 2e-16 ***
## Amount.Requested              1.509e-04  6.764e-06  22.310  < 2e-16 ***
## Monthly.Income               -2.275e-05  1.203e-05  -1.892 0.058675 .
## Open.CREDIT.Lines            -4.414e-02  1.024e-02  -4.311  1.7e-05 ***
## Inquiries.in.the.Last.6.Months  3.504e-01  3.664e-02   9.564  < 2e-16 ***
## fico                         -8.562e-02  1.287e-03 -66.547  < 2e-16 ***
## Loan.Length_36months         -3.233e+00  1.180e-01 -27.388  < 2e-16 ***
## State_TX                      6.253e-01  1.769e-01   3.536 0.000415 ***
## Home.Ownership_MORTGAGE      -2.623e-01  9.238e-02  -2.839 0.004566 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.061 on 2190 degrees of freedom
## Multiple R-squared:  0.7571, Adjusted R-squared:  0.7562
## F-statistic: 853.4 on 8 and 2190 DF,  p-value: < 2.2e-16
```

We can use this model to make prediction on the test data

```
pred.IR=predict(fit.final,newdata=ld_test)
```

If we need , we can write this to a csv file as well. Notice, since we do not have response values for test data, there is no way for us to calcualte any sort of performance measure for it beforehand. We can only make a claim that performance might be roughly same as validation data.

```
write.csv(pred.IR,"mysubmission.csv",row.names = F)
```

Interpreting Model Coefficients

There are many other observation you can make from the coefficients. For example :

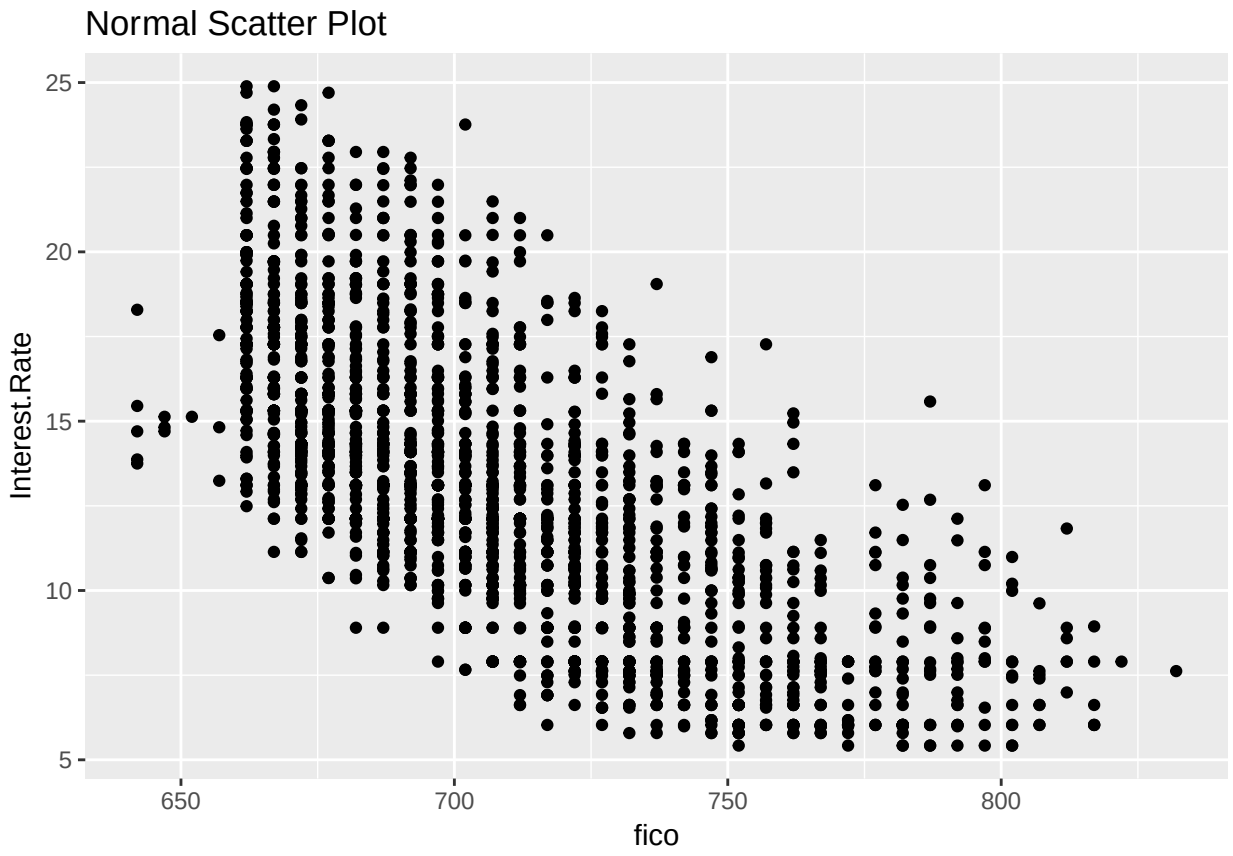
- For high amount requested , interest rates are going to be higher as well
- If someone has been applying for loan at too many places [results in high number of inquiries], they'll get higher interest rates
- Where as if you have high fico score [mroe credit worthy] , your interest rates are going to be lower.
- If you are applying for a loan for 36 months , your interest will be lower by almost 3.5% in comparisons to 60 months loan

with these and many more observations , you can come up with good recommendations for your clients.

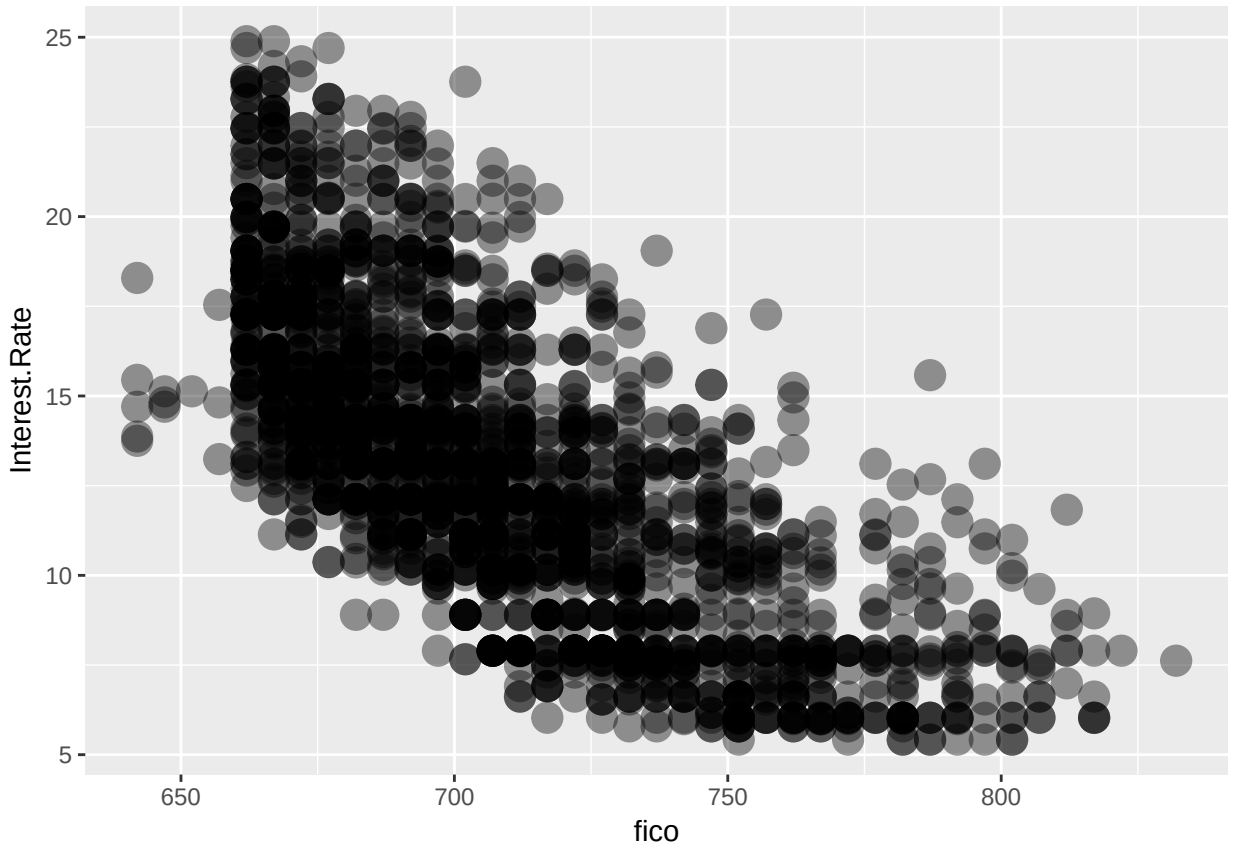
Remember , what we went through here is one of the possible iterations. You can experiment with your own variable transformation etc and come up with different models which might perform better.

Extra Notes on Scatter Plots For Large Data

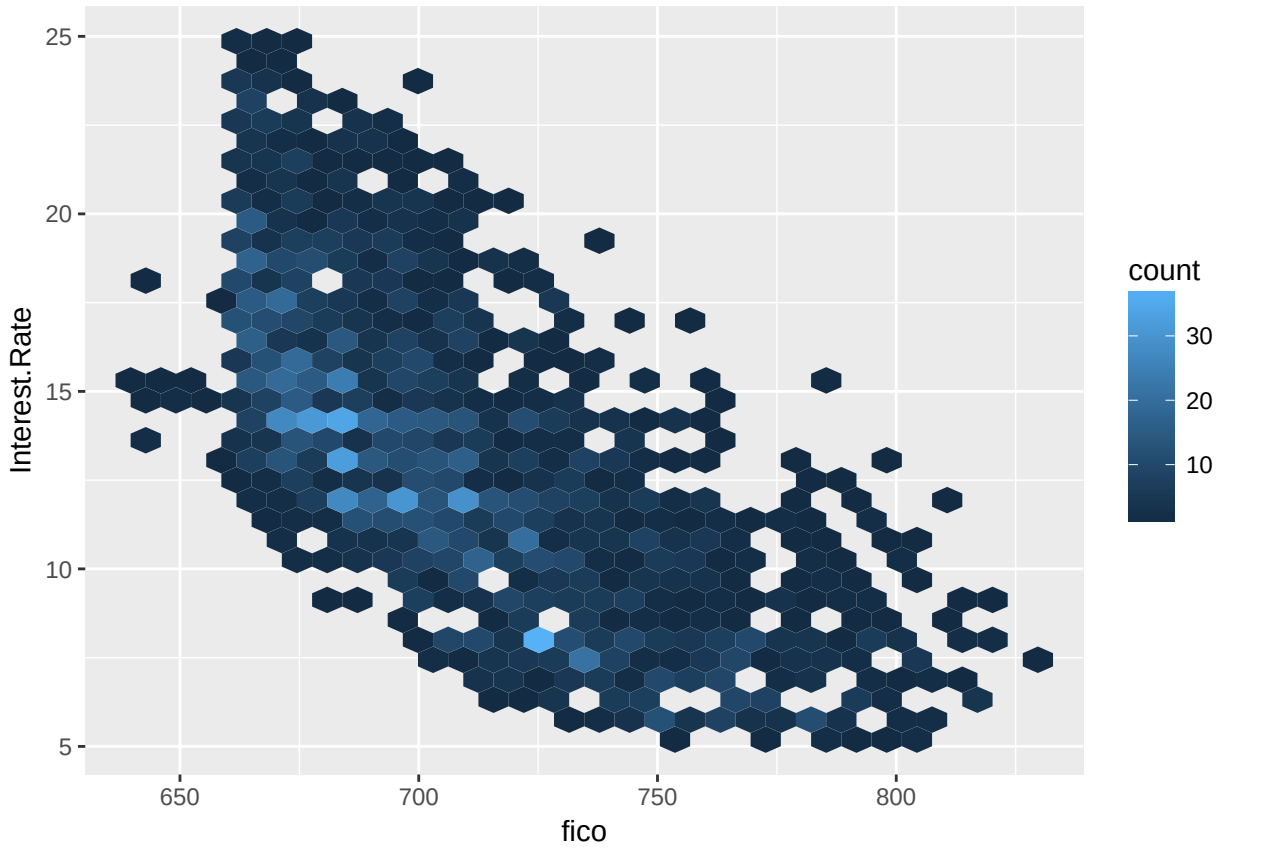
We have seen that as datasets get large , scatter plots start to look like blobs and not much can be deciphered. This happens due to overlapping of points. There is a way we can see what part of the scatter plots are dense and focus on only dense parts to identify a pattern if there exists one. Here are few ways to do the same.



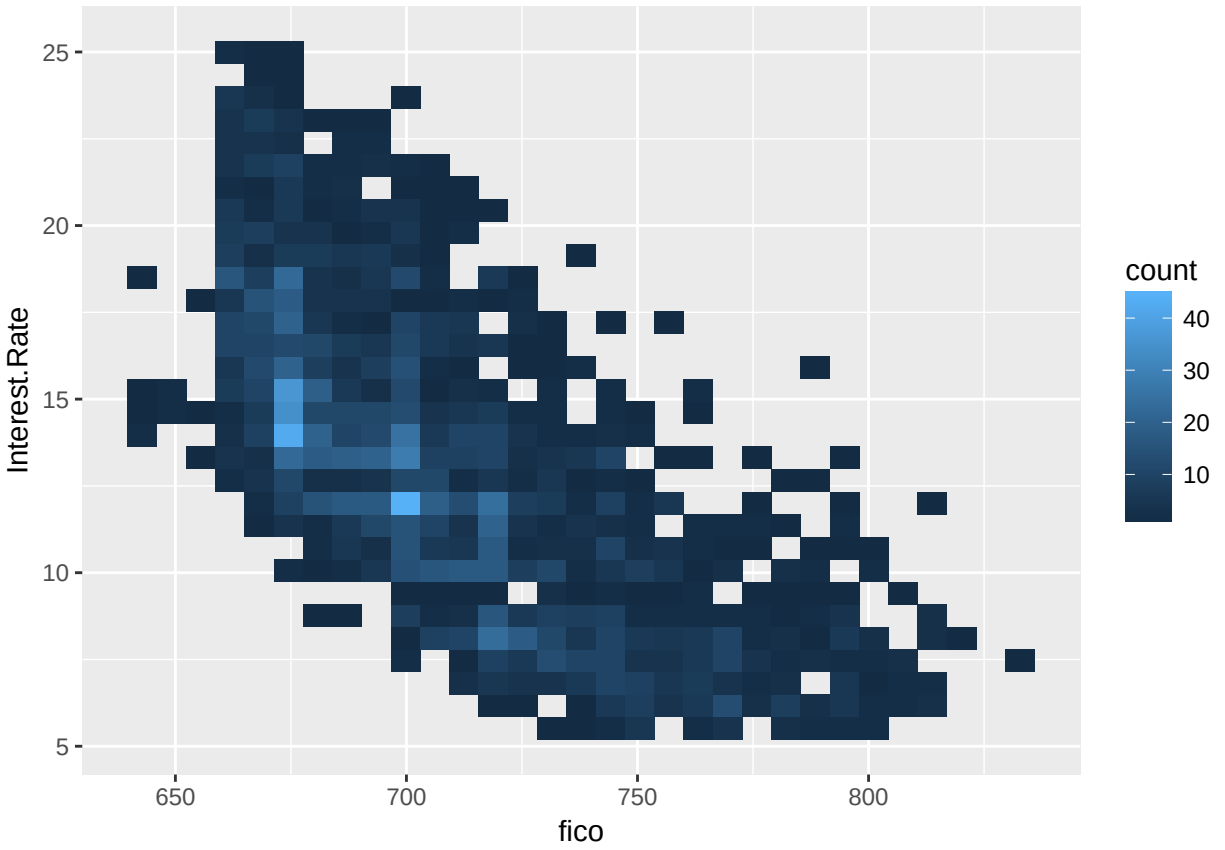

```
# Scatter Plot with density mapped
ggplot(ld_temp,aes(x=fico,y=Interest.Rate))+geom_point(alpha=0.4,size=5)
```




```
#scatter plot with hex binning or 2d binning , you'll need to install package "hexbin" for the same
ggplot(ld_temp,aes(x=fico,y=Interest.Rate))+stat_binhex()
```




```
ggplot(ld_temp,aes(x=fico,y=Interest.Rate))+stat_bin2d()
```



We'll conlude here. You can play around with linear regression model with the other datasets shared with you. Contact if you face any issue.

Prepared By : Lalit Sachan

Contact : lalit.sachan@edvancer.in

In case of any doubts or clarification please take to QA forum on LMS