Data Mining Coursework

By

Baseer Ahmad

UOB number

18018515

University of Bradford

United Kingdom

Table of Contents

[Task 1 Data Choice 3](#_Toc8207866)

[1.1 The train dataset 3](#_Toc8207867)

[1.2 The test dataset 3](#_Toc8207868)

[Task 2 Data Analysis 5](#_Toc8207869)

[2.1 Introduction 5](#_Toc8207870)

[2.2 Step 1: uploading the dataset 5](#_Toc8207871)

[2.3 step 2: Checking the structure, summery, Mean average, median and Mode of the data 5](#_Toc8207872)

[2.4 Step 3: simple bar plot for finding how many passengers are in the ship 6](#_Toc8207873)

[2.5 Step 4: Checking Missing Data and values 8](#_Toc8207874)

[2.6 Step 5: Exploratory Data Analysis on survival of passangers 10](#_Toc8207875)

[Task 3 Classification of Data 13](#_Toc8207876)

[3.1 Decision tree 13](#_Toc8207877)

[3.2 Pros Decision Trees: 13](#_Toc8207878)

[3.3 Cons Decision Trees: 13](#_Toc8207879)

[3.4 Step 1: Splitting datasets 14](#_Toc8207880)

[3.5 Step 2: Fitting Decision Tree Classification Model to the Training set and tree visualization 14](#_Toc8207881)

[3.6 Step 3: checking prediction accuracy 15](#_Toc8207882)

[3.7 Step 4: Applying k-Fold Cross Validation 16](#_Toc8207883)

[Task 4 Text Mining 18](#_Toc8207884)

[4.1 Step1: dataset uploading 18](#_Toc8207885)

[4.2 Step 2: download the English model and take 1-year data for processing 19](#_Toc8207886)

[4.3 Step 3: Further filtering part of speech 20](#_Toc8207887)

[4.4 Step 4: Word Cloud 24](#_Toc8207888)

[Task 5 Critical Review 28](#_Toc8207889)

# Data Choice

On 1912, 15 April, the RMS Titanic British passenger ship which was go down in the early morning in North Atlantic Sea after hitting by means of an iceberg all through her earliest journey beginning from Southampton to New York City. There are 2224 crew and passenger embarked. There are 1500 passengers died during the unlikely event occurs.

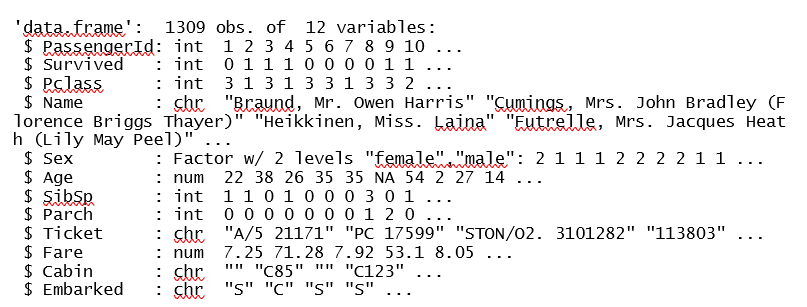
the Titanic data set obtained from Kaggle.com and I have analyzed it for this course work. Using the data set, individual container perceive that 891 travelers of the Titanic perished or survived. The other analysis term is the relationship between more than a few variables, as well as Sex , Age , passenger class, if they had family on-board the ship, their ticket number, how much they paid for their ticket, where they boarded the ship, and their cabin’s location. To tackle this problem, I will apply the conditional inference tree algorithm to train classification models of survival using several of the passengers’ behaviors. The resulting models will be evaluated using balanced accuracy for my own validation data set, and ultimately, I will be submitting the model predictions to the Kaggle competition’s public leaderboard and receiving a score. The available data are spliced in to two group the Test dataset and train dataset

## The train dataset

This dataset is given for using to train our model for machine learning as training dataset. We can make our model on “features” based like passenger class and gender. We can also create new features by using feature engineering.

## The test dataset

This dataset is given for testing our model to perform some prediction on the passenger survival rate. And look at the result how our model performs the prediction task on it and what is the accuracy of our model when we use this dataset. We use the test dataset to check the probability of each passenger that survived or not survived during the titanic sunk, as well as with respect to their passenger class.



Both datasets are combined in the data analysis task to perform some data analysis on do some graphical representation of each variable. The combined dataset contains 12 variables and 1309 observations as shown in the above figure. Which is the following.

1. Passenger id is integer type variable
2. Survived is integer type variable and contain 1 = Yes, 0 = No
3. Passengers class (Pclass) is integer and variable and contains ticket class 1 = 1st, 2 = 2nd and 3 = 3rd class
4. Names of passengers is a character type variable
5. Sex is Factor type variable and contains gender information Male/ Female
6. Age is number type variable and contains age in years
7. Siblings (SibSp) is integer type variable contains information about Sibling = sister ,brother, stepsister , stepbrother, and Spouse = wife , husband
8. Parch (families relatives ) is integer type variable contains information about family relationships
9. Tickets is a character type variable
10. Fare is number type variable
11. Cabin is a character type variable
12. Embarked is a character type variable and contain port information C = Cherbourg, Q = Queenstown, S = Southampton

The second dataset I choose for the text mining task is also taken from Kaggle.com. the chosen dataset contains information about 15 years of headlines. This dataset discussed in more detail in the text mining task.

# Data Analysis

## Introduction

In this task I analysis the Titanic dataset and prepare it for prediction of the Titanic passenger survival chances which I will do in the next classification task. To approach this problem of defining who perished or survived in the tragedy of Titanic accident, I combine the train dataset and test dataset to do the exploratory analysis. The analysis consisted of plotting different variables of the Survived field to check the relationships of the different passengers’ classes. I simply use three variables: Sex, Passenger Class (Pclass) and Age.

The following questions are addressed in this coursework:

1. Prediction for entire ship survival.
2. What is the connection between a passenger’s chance of survival and features?

Now, further explore the data I use R studio to upload the dataset and get a brief summary of each variable.

## **Step 1**: uploading the dataset

set the working directory by using the following command.

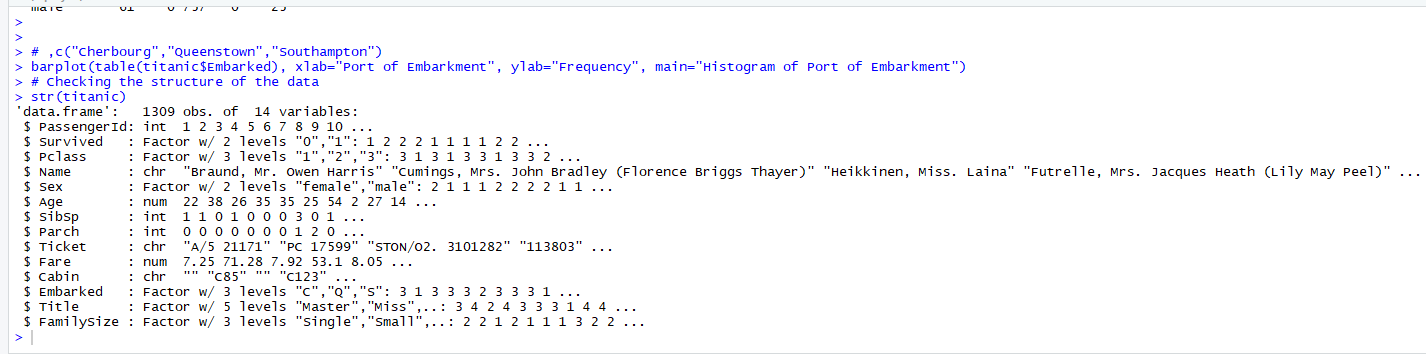
* setwd("D:/Rproject")
* getwd()

Then upload the dataset and combine it for further analysis

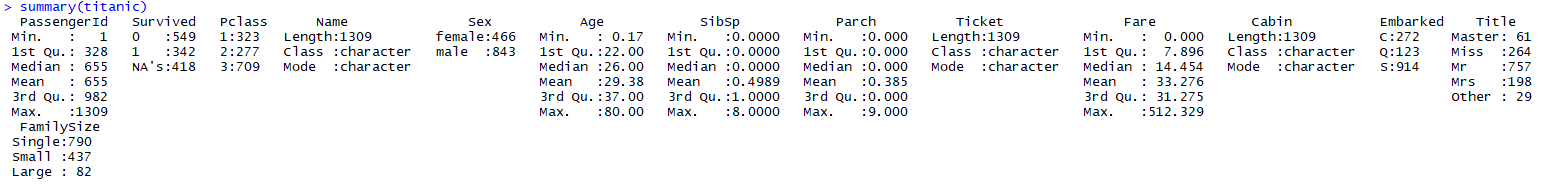
* titanic\_train\_dataset = read.csv("titanic\_train\_dataset.csv")
* titanic\_test\_dataset = read.csv(' titanic\_test\_dataset.csv')
* titanic\_full\_dataset <- bind\_rows(titanic\_train\_dataset, titanic\_test\_dataset)

## **step 2:** Checking the structure, summery, Mean average, median and Mode of the data

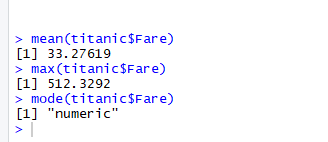
* str(titanic\_full\_dataset)
* summary(titanic\_full\_dataset)
* mean(titanic\_full\_dataset $Fare)
* max(titanic\_full\_dataset $Fare)
* mode(titanic\_full\_dataset $Fare)



In the structure we see that, there is 1309 observation of 14 variables. As we discussed in task 1.

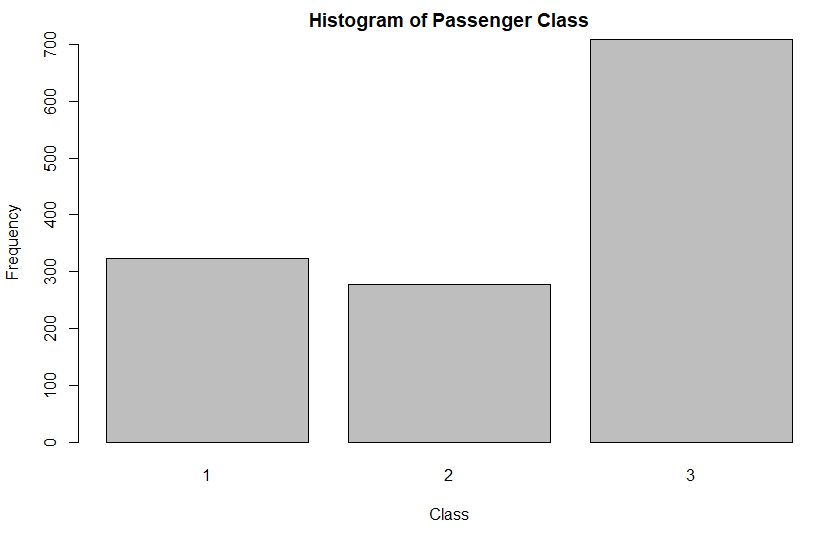


In the summery we see the details about the 14 variables. There family size, male/ females, embarked from three ports etc.

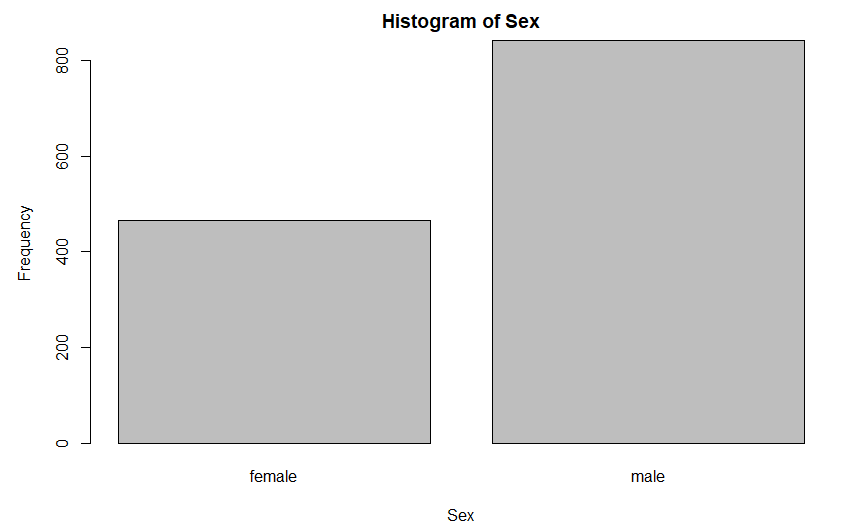


By using mean, max, mode and median by using above commands to see the results of every field.

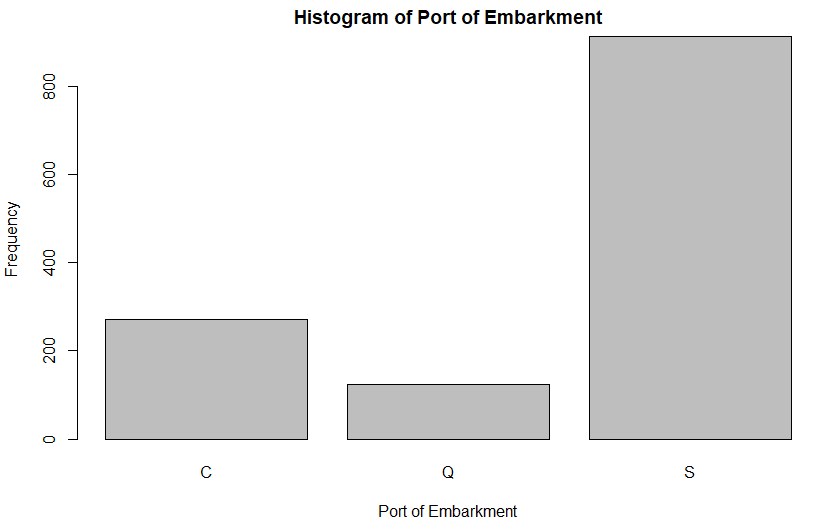
## **Step 3:** simple bar plot for finding how many passengers are in the ship

* barplot2(table(titanic\_full\_dataset$Pclass), main="Barplot of Passenger Class",xlab="Class", ylab="Number of Passengers ", col = "red")
* barplot2(table(titanic\_full\_dataset$Sex) , main="Barplot of Sex", xlab="Sex", ylab="Number of Passengers ", col = "red")
* barplot2(table(titanic\_full\_dataset$Age) , main="Barplot of Age", xlab="Age", ylab="Number of Passengers ", col = "red")
* barplot2(table(titanic\_full\_dataset$Embarked) , main="Barplot of Port of Embarkment" , xlab="Port of Embarkment", ylab="Number of Passengers ", col = "red")

In this plot we see that there are three passenger classes and most of the passengers are in the 3rd class .



The above plot shows that most of the passengers are male.



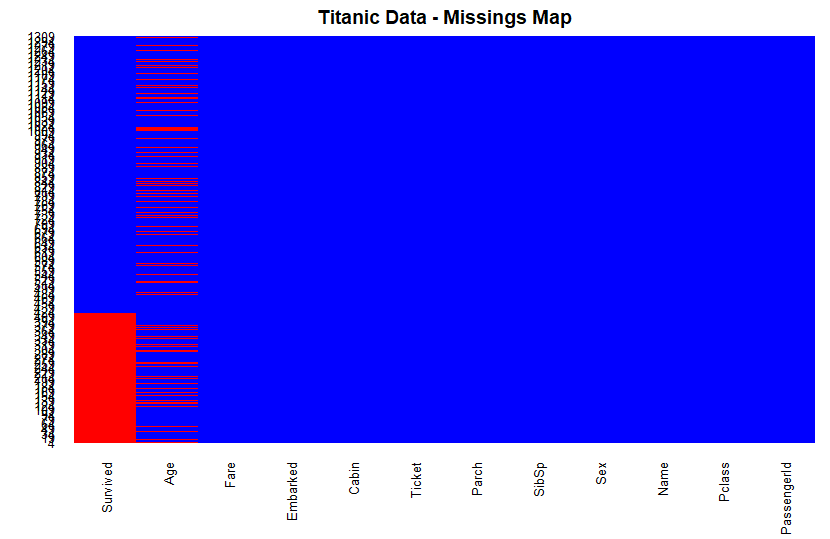
This graph shows that most of passengers embarked from Southampton port ,second from Cherbourg and third from Queenstown.

## **Step 4:** Checking Missing Data and values

* colSums(is.na(titanic\_full\_dataset)|titanic\_full\_dataset=='')
* missmap(titanic\_full\_dataset, main="Titanic Data - Missings Map", col=c("red", "blue"), legend=FALSE)



In the above figure we noticed that the missing values are in survived = 418, cabin = 1024 , Age = 263, Fare = 1 and Embarked = 2 below is the graphical representation of this data.



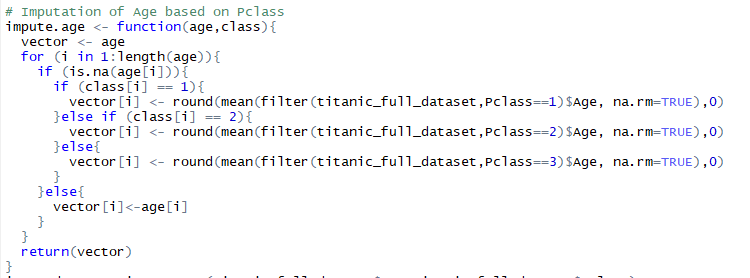
Now Extract the row which contains the missing Fare and impute it from the median from 3rd class values

* filter(titanic\_full\_dataset, is.na(Fare)==TRUE | Fare=='')
* titanic\_full\_dataset$Fare[is.na(titanic\_full\_dataset$Fare)==TRUE] = median(filter(titanic\_full\_dataset, Pclass==3 & Embarked=="S")$Fare, na.rm=TRUE)

now extract the row which contains the missing Embarked values and also check the number of passengers from 1st class and fill the value from Cherbourg port

* filter(titanic\_full\_dataset, is.na(Embarked)==TRUE|Embarked=='')
* table(filter(titanic\_full\_dataset, Pclass==1)$Embarked)
* titanic\_full\_dataset$Embarked[titanic\_full\_dataset$Embarked==""] = "C"

finally impute the missing value of Age on Pclass by using the following function and then check the result for missing values



* imputed.age <- impute.age(titanic\_full\_dataset$Age,titanic\_full\_dataset$Pclass)
* titanic\_full\_dataset$Age <- imputed.age
* colSums(is.na(titanic\_full\_dataset)|titanic\_full\_dataset=='')



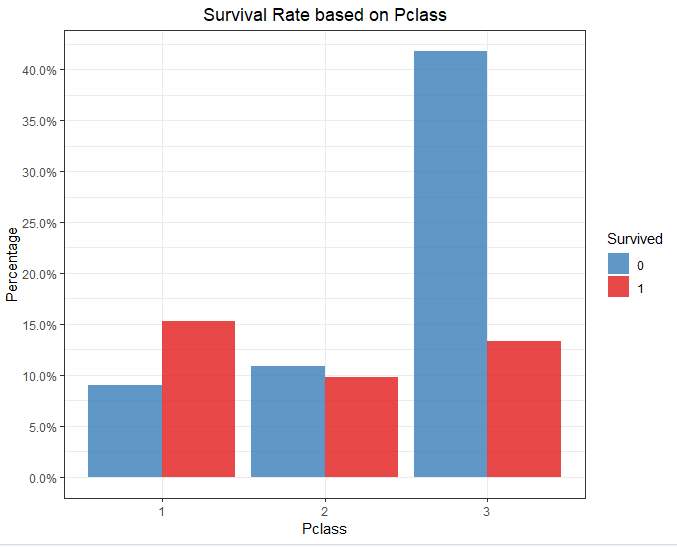
## **Step 5:** Exploratory Data Analysis on survival of passangers

Now encoding the categorical features as factors and then check the structure of data

* titanic\_full\_dataset$Survived = factor(titanic\_full\_dataset$Survived)
* titanic\_full\_dataset$Pclass = factor(titanic\_full\_dataset$Pclass)
* titanic\_full\_dataset$Sex = factor(titanic\_full\_dataset$Sex)
* titanic\_full\_dataset$Embarked = factor(titanic\_full\_dataset$Embarked)
* titanic\_full\_dataset$Title = factor(titanic\_full\_dataset$Title)
* titanic\_full\_dataset$FamilySize = factor(titanic\_full\_dataset$FamilySize, levels=c("Single","Small","Large"))
* str(titanic)

Now do some graphical representation of Pclass, Sex and Age survived and non-survived passenger in each class

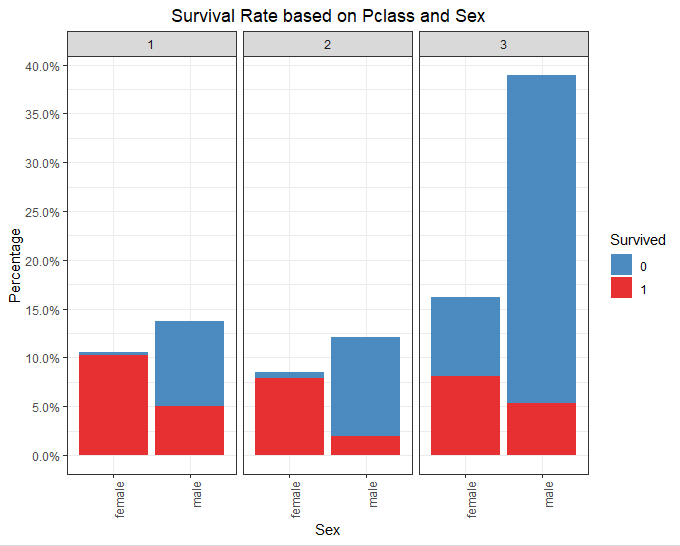
* ggplot(filter(titanic\_full\_dataset, is.na(Survived)==FALSE), aes(Pclass, fill=Survived)) + geom\_bar(aes(y = (..count..)/sum(..count..)), alpha=.8, position="dodge") + scale\_fill\_brewer(palette = "Set1", direction = -1) + scale\_y\_continuous(labels=percent, breaks=seq(0,0.6,0.05)) + ylab("Percentage") + ggtitle("Survival Rate based on Pclass") + theme\_bw() + theme(plot.title = element\_text(hjust = 0.5))



The above graph shows that the survival ratio of class 1 is 15%(red bar), in class 2 its 10% and in class 3 its 12.5%.

Now we show the same result with Pclass and Sex

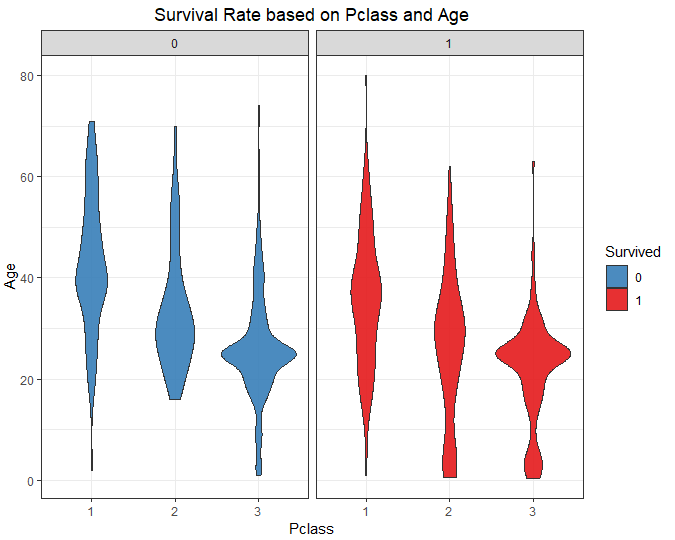
* ggplot(filter(titanic\_full\_dataset, is.na(Survived)==FALSE), aes(Sex, fill=Survived)) + geom\_bar(aes(y = (..count..)/sum(..count..)), alpha=0.9) + facet\_wrap(~Pclass) + scale\_fill\_brewer(palette = "Set1", direction = -1) + scale\_y\_continuous(labels=percent, breaks=seq(0,0.4,0.05)) + ylab("Percentage") + ggtitle("Survival Rate based on Pclass and Sex") + theme\_bw() + theme(plot.title = element\_text(hjust = 0.5)) + theme(axis.text.x = element\_text(angle = 90, hjust = 1))



The above graph shows that most females survived in 1st class whish is 10% , 8% in second and 3rd class and less male are survived than female.

Now finally we show the result with Pclass and Age

* ggplot(filter(titanic\_full\_dataset, is.na(Survived)==FALSE), aes(Pclass, Age)) + geom\_violin(aes(fill=Survived), alpha=0.9) + facet\_wrap(~Survived) + scale\_fill\_brewer(palette = "Set1", direction = -1) + ggtitle("Survival Rate based on Pclass and Age") + theme\_bw() + theme(plot.title = element\_text(hjust = 0.5))



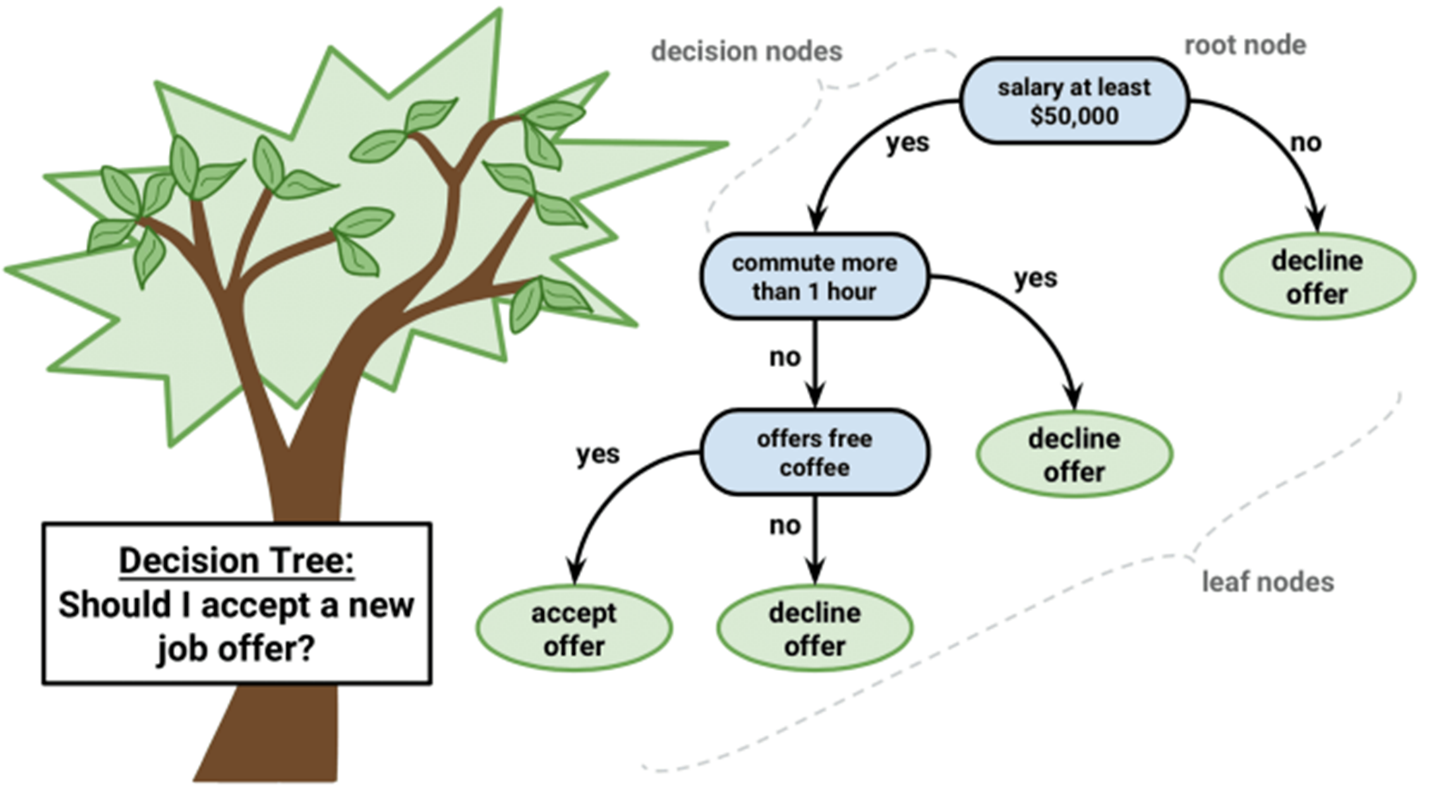
The above class is the survival ratio of different ages in passenger class where most of survived persons are in 3rd class and age between 20 to 30 years old. And in class 1st 30 to 40 years old

# Classification of Data

In this task I continue the code from analysis task to predicate survival of the Titanic passengers using decision tree classification techniques. In order to do this, I slit the data in to test and train datasets.

## Decision tree

The reason behind choosing the decision tree is that, it’s a graphical illustration of all the probable resolutions to a conclusion based on confident circumstances. The decision tree starts from single box and grow up different boxes to show different possible solutions. The decision tree is designed to work with different classes of analysts. All kind of definite variables didn’t have any actuall problem by means of decision trees. Business can make decision easily with model gotten from decision tree. Due to highly biased class of models any biggest problem can be solved with decision tree. The model can be made from training datasets.



## Pros Decision Trees:

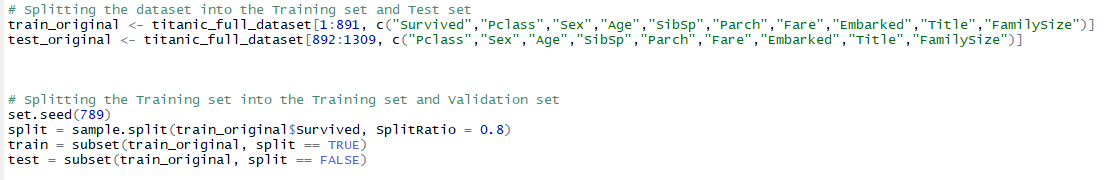
* Instinctual Decision Rules
* Take into account variable interactions
* non-linear features Can be handled easily

## Cons Decision Trees:

* No ranking score as direct result
* Highly biased to training set

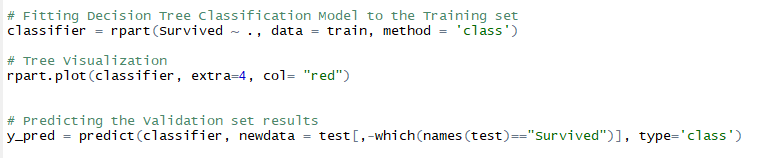
Now we start the classification of dataset by splitting it into train and test datasets

## **Step 1:** Splitting datasets

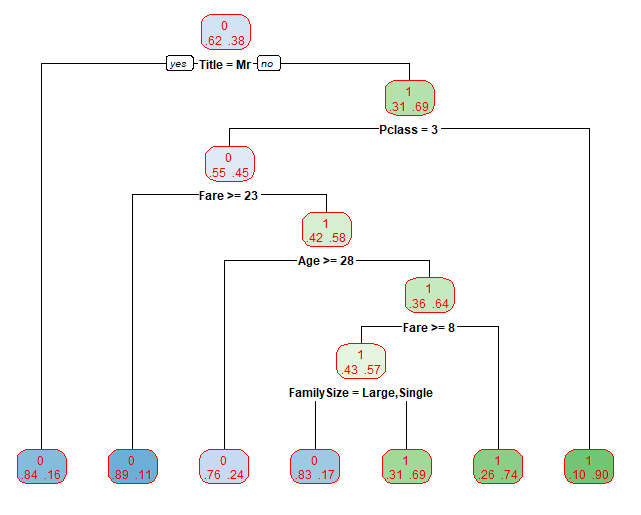


In the above command I choose the dataset rows from 1 to 891 for training and from 892 to 1309 for testing. Then by using **set**.**seed()** command which is suitable for creating random objects that can remain repeated. Then by using **sample.split()** command which split the data with certain ratio I choose .8 percent from the entire dataset.

## Step 2: Fitting Decision Tree Classification Model to the Training set and tree visualization

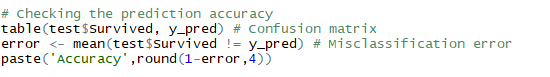


In the above commands I use **rpart()** function the first part of the function is a model formula by means of the ∼ symbol known as “is modeled as”. For visualization of decision tree I use the **rpart.plot()** function in which first argument is the classifier which we take from previous command and the next argument is “**extra**” it shows some extra information of the class probability on each node it can be set “auto” or choose from 0 to 106 with different values. The detail information is available here (rplot, 2019). And then we set the predicting validation set results by using predict function. The first argument is the classifier and second is the survived data from test class .

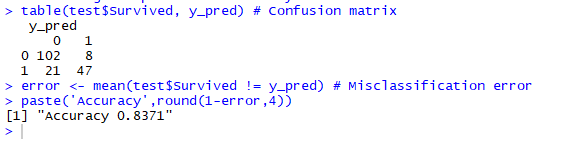


The above at the top decision tree root shows that 62% percent of passenger will die while 38% percent will survive .now move to the next branch of the tree which shows that if with “Mr” title 16% have chance of survival while 84% will die , now move to the left side of the tree on second node which show the relation of passenger class has 69% of non “Mr” title persons are survived while 31percent will die. Further down if the passenger class is 3 then the chance of survival is 90% if not then the chance of survival is 45%. Further more if Fare is greater then 22 survival chance is 64% percent while other has 11% percent, further down if the age is greater than 27 survival chance is 64% percent while less than 28 is 24% and the tree is go down further and show the detail of the testing datasets against the training datasets.

## Step 3: checking prediction accuracy



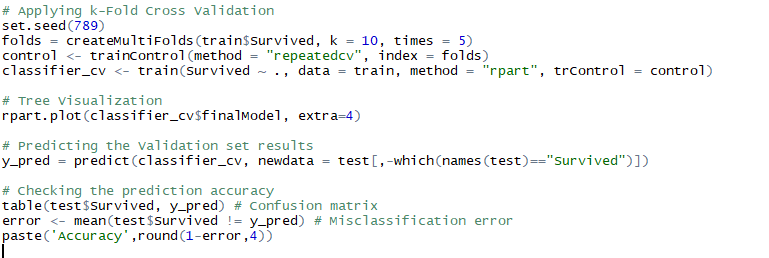
By using the above code, we find the accuracy of the prediction using confusion matrix method.



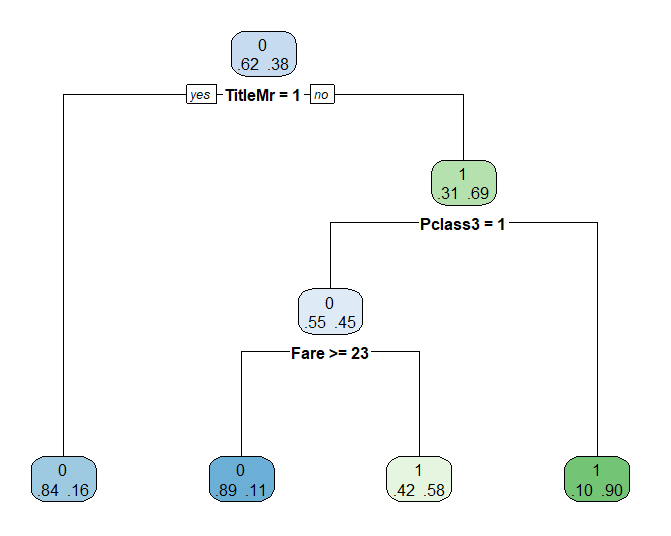
The above result shows the confusion matrix and the accuracy of our model which is 83 percent.

## Step 4: Applying k-Fold Cross Validation

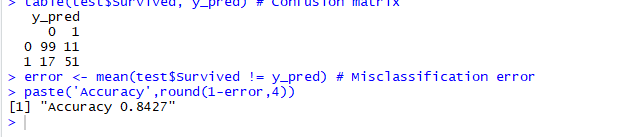
In this technique I divided the dataset in 10 groups of almost equal size. The first group is use for testing and the other groups were used to train the model. The k Fold cross validation give us more efficient performance model.



In the above code I use the set.seed function for simulation of randomly generated number of times. And make groups called fold by using the **creatMultifolds()** function. And then set the train clasifer and then plot the efficient decision tree. After making the decision tree we check the accuracy of the model which is 84 percent this time.



In the above figure we see that the top root of the decision tree show that 38% percent of the passenger will survive while 62% will die. “Mr” title persons have 84 percent of chance that are not survived than other name titles. The current decision tree is more precise than the previous decision tree and has better accuracy.



Here is the accuracy of 84% of the k fold model.

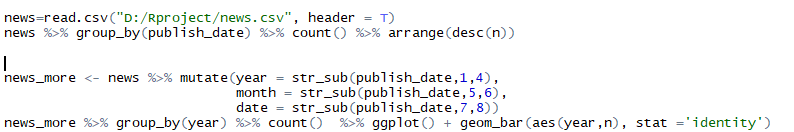
# Text Mining

The text mining is a technique which is highlight the most frequent words used in text or a paragraph. In this method we make a graphical representation of many frequent used words to understand it easily. For example, if we want to compare two different speeches of the same person on different or same topics. After comparison and make a graphical representation of the model we can easily conclude the aim of that speeches.

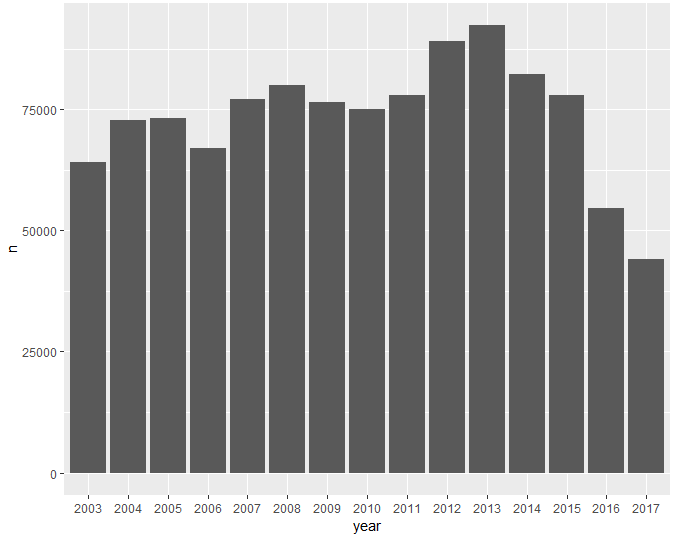
To perform this task, I choose the dataset from [kaggle.com](https://www.kaggle.com/therohk/million-headlines/downloads/million-headlines.zip/7). the dataset contains the 15 years headlines from newspapers of ABC (Australian Broadcasting Corp.). the aim and outcome of the dataset is to find out most keyword used in 15 years which help us to find out the conclusion of the newspaper over the years and the reports which published in the paper in top stories of the year. The data can be representing in graphical form in which we take word from Part of speech and the finally at the end we make a word cloud to make a better graphical representation of the data.

## Step1: dataset uploading

To start the process first we need to upload the dataset csv file and set it into descending order



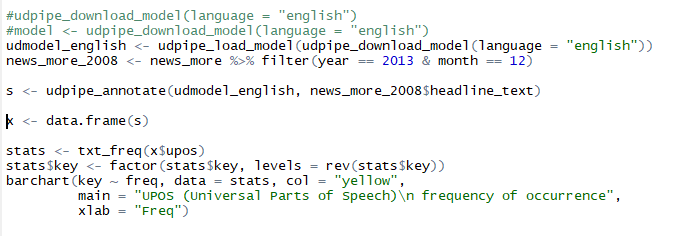
After this we need to further analyze the data and check for the year in which most of the headlines are published. For this first we separate the data by dates and year. And then plot the data.



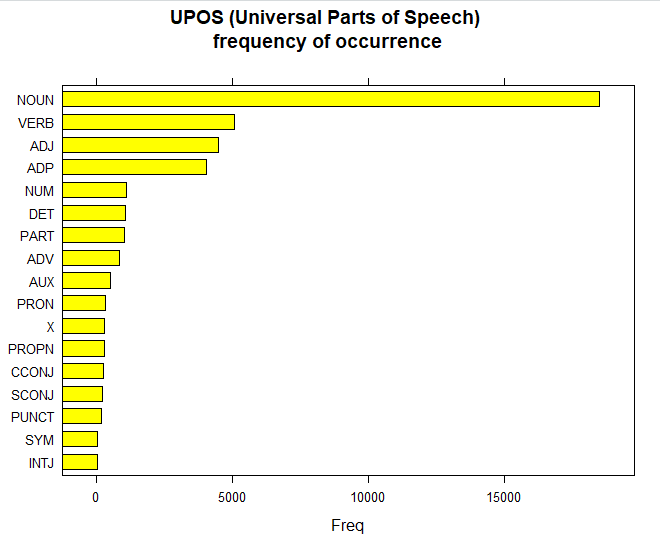
According to the above graph in 2013 most of the headline are published in the newspapers.

## Step 2: download the English model and take 1-year data for processing

After finding the graph of published headlines in newspaper then I download the English ud library to further search for words and part of speech data in to the dataset

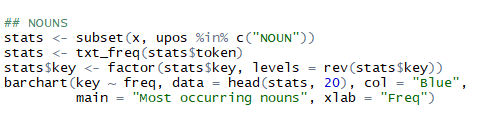


In the above code I select the 2013 and 2014 data for further processing, then make data frame of two years data for filtering the words and frequency of common words.

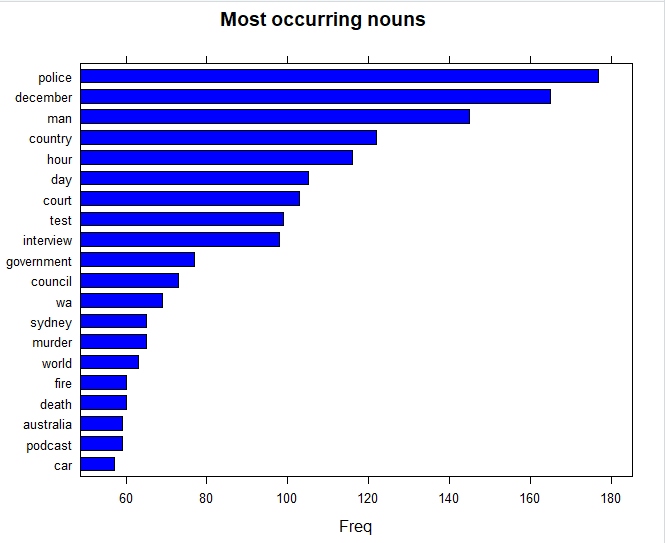


After plotting the data frame, we noted that most of the words are “NOUN” with frequency of more than 15000 words.

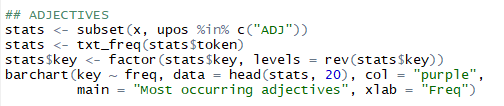
## Step 3: Further filtering part of speech



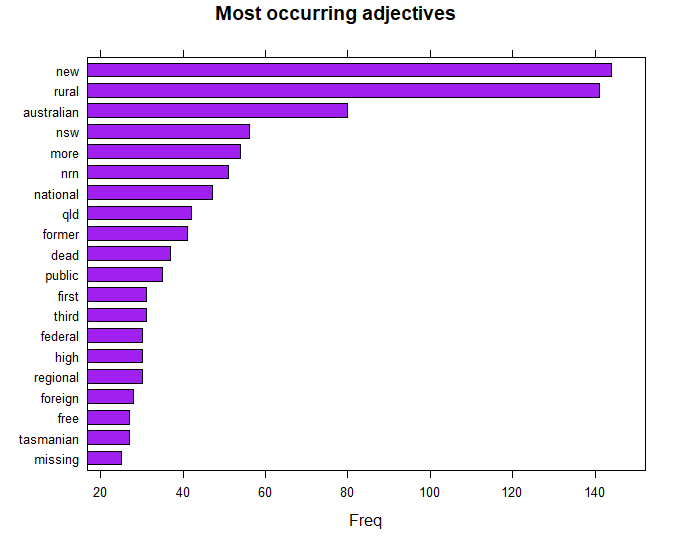
By using the above code, we search for only NOUNS in the text



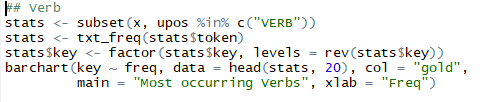
In the above figure we noted that the word “police” is used more than 170 times in the 2 years data frame. Then December and so on.



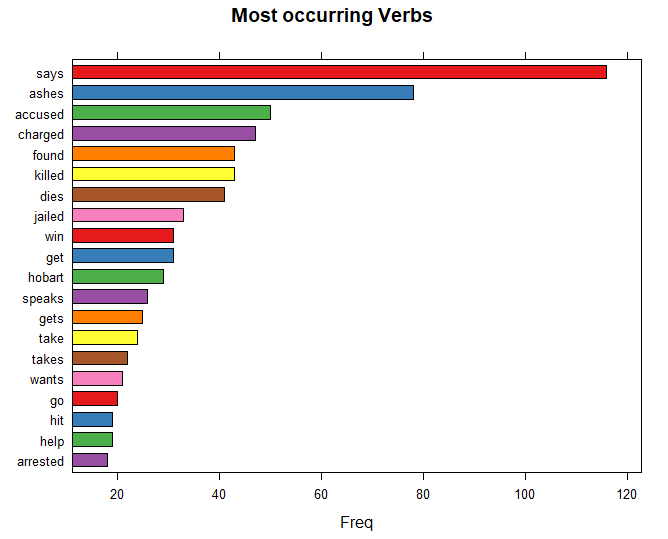
By using the above code, we further filter the most used “Adjective” in the news paper are shown in below figure.



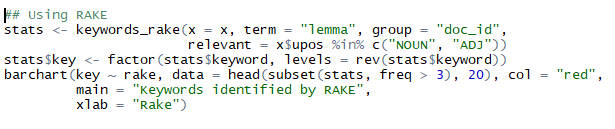
The word “new” is used more than 140 times in the 2 years data frame.

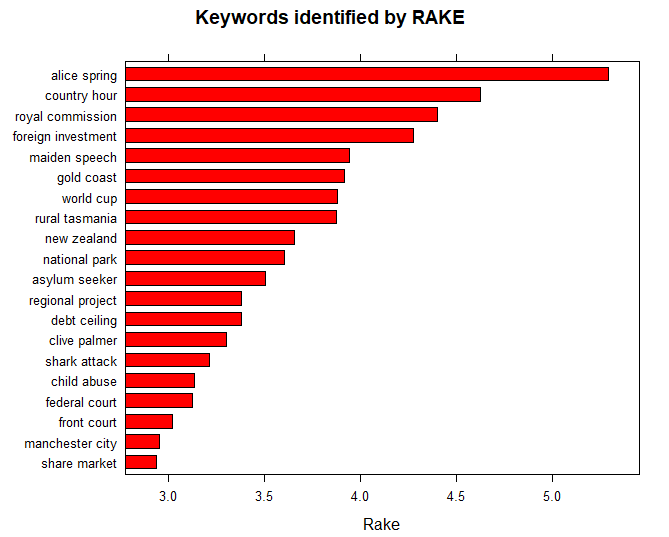


In the above code we filter the most “VERB” used in the paper are

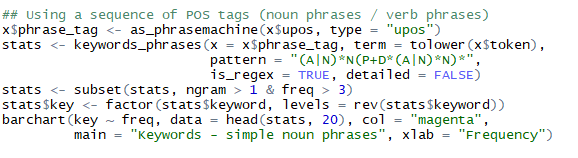


In the above figure most occurring verb is “Says” which is more often used in the newspaper and frequency of that word is more than 110.

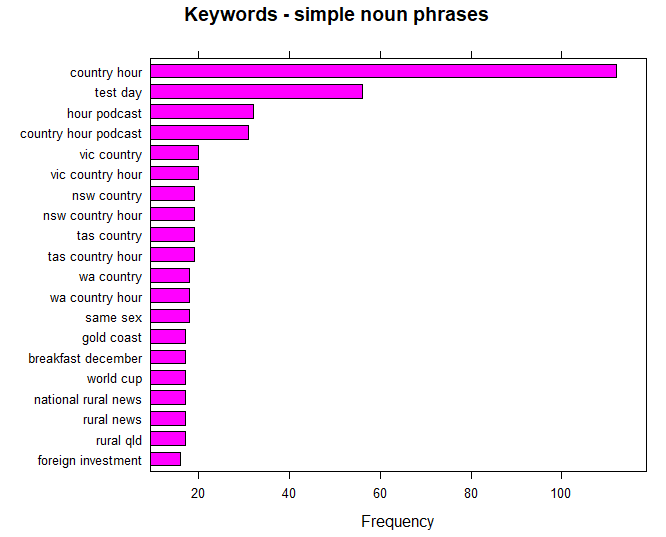


In the above code I filter the Rake data which means the NOUN and ADJ used together in a sentence.

In the above graph the most common words used together are “Alice spring” which is mostly used with nouns.



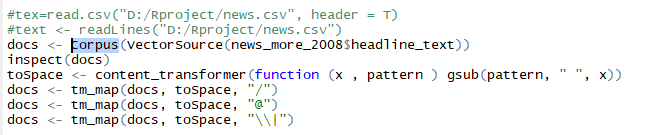
Now finally in above code we filter simple noun phrases which most commonly used in sentences and the most common words “country hour” are used more than 100 times in the newspaper data frame .



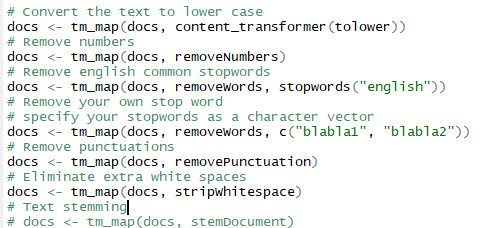
## Step 4: Word Cloud

In the previous steps we just find out different common words on to understand and quickly read and analyze all the common words at one time we need to collect and display it at one place. The technique used to perform this step is called word cloud.

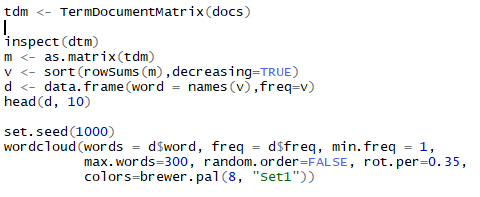
To start the process first we need to load the file using Corpus() function which is the list of documents in our case we use only one document.



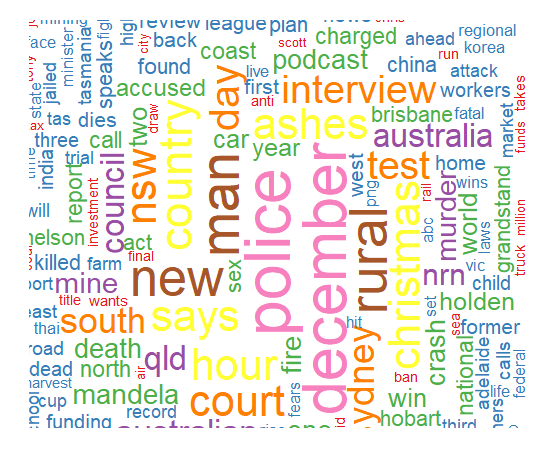
In the above code we load the data and then inspect the data to further see the data type and other data types. After inspecting we remove all unwanted symbols and signs from the text and left only text characters. To remove the extra symbols, I used the tm\_map() function from text mining package.



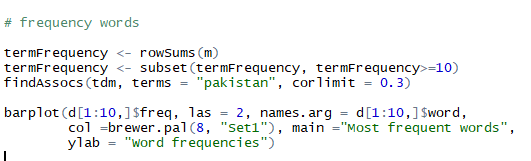
In the above code I further remove white spaces, numbers, stop English common word like (the, a, an etc. ) and convert all text to lower case. With removeNumbers , and removePunnctuation arguments and so on. To reduce words to their root forms, make text stemming as important part of preprocessing the text data. In simple words, to get the common origin this process removes suffixes from words.



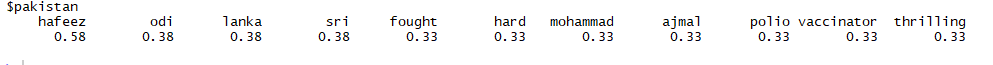
In the above code I make a document data frame called tdm (term document Matrix ) . it contains the common frequency words. Its row contains documents and column names are words. Then make a data frame for word cloud and show the words by using the wordcloud() function in which “min.freq” argument means show the number at least 1 time because I choose frequency equal to 1. If we choose 10 then it means it will show the result of those words which appeared at least ten times in the selected year data frame. The second important argument is “max.word” which means how many words do you want to show in result.



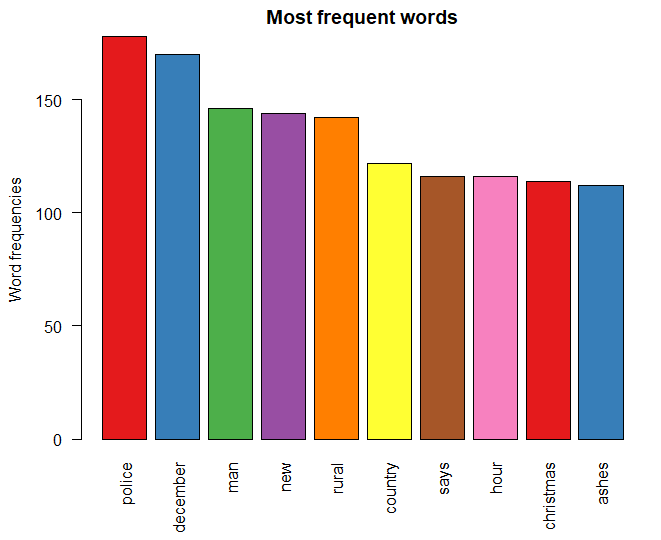
Here is the final result from the word cloud function which look great. And everyone can easily understand the in 2013 to 2014 more headlines are based on criminal activities because the police word is most commonly used in the news paper . every word is appearing with different color according to their frequency.



Now finally by using the above code we find the relationship of different words with other word with 30% percent ratio. By using findAssocs() function. And then display the common word on a bar graph .



In the above figure the word “Pakistan” has relationship with different word like with “hafeez” its 58% now I am from Pakistan , so I quickly imagen that there is an ODI cricket between siri Lanka and Pakistan and Hafeez are playing that match. In conclusion every person can easily analyze the graphical text data.



In the above figure the most common word is Police as we discus it in the Word cloud graph .

# Critical Review

In the entire coursework I use the r script because I am familiar with R studio and also, I have one ongoing project based on machine learning. R studio provide facilities of many library’s and datasets which is already available in R studio. The titanic dataset I chosen is also available in r, the idea for selection is came up from R using “dataset::” command.

However, there are some difficulties I have faced during the course work like I started the association rule mining task, everything’s goes well until apriori(titanic.data) algorithms applied and when I want to visualize the data the required the “arulesViz” library I installed multiple time even reinstall the r studio but still the same problem then I choose the classification task.

The conclusion of the entire course work is that I really enjoy the text mining task, because I try different dataset including my sms conversations and find really help full when analyzing large dataset of text data, the text mining application will be used in checking exam paper to analyze the progress of the students. Teachers can easily take feedback from the text mining applications. In future I want to work on shinny web app which is also useful tool of r studio I find out many tutorial where people just put the world cloud and other graphical code which I used in this coursework inside the shiny web app and export it as application. The user just needs to put plain text file and they can select the minimum frequency of word from the list available on the dashboard.

In the data analysis and classification task I want to use more algorithms in future to perform the same task using different techniques like Logistic regression, SVM and Random Forests techniques. The best thing about the random forest technique I learned from different tutorials is they create many decision trees using bootstrapping which increase the prediction accuracy more. The second thing I want to use ARIMA based models for prediction and find out its Mean Absolute Percentage Error.