**PRACTICAL 5**

AIM: Implementation of the basics of R and data acquisition, Install packages , Loading packages Data types, checking type of variable, printing variable and objects (Vector, Matrix, List, Factor, Data frame, Table) cbind-ing and rbind-ing, Reading and Writing data. setwd(), getwd(), data(), rm(), Attaching and Detaching data. Reading data from the consol. Loading data from different Data sources.(CSV,Excel).

THEORY:

Basic R commands:

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool. It was designed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

Features of R:

Statistical features:

Basic Statistics: The most common basic statistics terms are the mean, mode, and median. These are all known as “Measures of Central Tendency.” So using the R language we can measure central tendency very easily.

Static graphics: R is rich with facilities for creating and developing interesting static graphics. R contains functionality for many plot types including graphic maps, mosaic plots, biplots, and the list goes on.

Probability distributions: Probability distributions play a vital role in statistics and by using R we can easily handle various types of probability distribution such as Binomial Distribution, Normal Distribution, Chi-squared Distribution and many more.

Data analysis: It provides a large, coherent and integrated collection of tools for data analysis.

Programming features:

R Packages: One of the major features of R is it has a wide availability of libraries. R has CRAN(Comprehensive R Archive Network), which is a repository holding more than 10, 0000 packages.

Distributed Computing: Distributed computing is a model in which components of a software system are shared among multiple computers to improve efficiency and performance. Two new packages ddR and multidplyr used for distributed programming in R were released in November 2015.

1. print() prints out code.
2. To assign a value to a variable, use the <- sign.

Description: https://lh3.googleusercontent.com/x3KeyqMpveTCMVbNb6KxzurZIHKFzAhBwcMpVVBqWx4zKJg2HqOrmjwyr2NGnYIzd5NcaRTyXJoozSsj0J4uMi3_qY0dLYCnSBXsEmPO3dzwdnPTEHU9D7yGTqXSTgxa4lrBGl0UmgrDwLZ5CvkRKzs

setwd(dir) is used to set the working directory to dir.

Description: https://lh5.googleusercontent.com/nQkBW07pVh_70ENnb7-D7nMr1Uwq5K6CV4wrTOKNq6K5T0f5hkTvnbiwizM_CirEb1e7UjW4rMO3juN5EDLcg8Fx5ayiLujCO3ZkyfZtDXM_swMrFc788zuNjGs6yatvbrYmhmGNrXlbHYHhA6SkbYQ

getwd(): It returns an absolute filepath representing the current working directory of the R process.

Description: https://lh5.googleusercontent.com/vVQSaiIZH0-RNNg85LJjnMJFgkEhZIDaDahYtkA51YX1syKdLBfrCJjxUUngVlSSWQOJ3aF6fgaTgOKuEzN5eaSW9B1T2LPZxPPu1Jkeg4cur8VFxiNfe2SoZRdm5hRCyBb2Tuc2b6H2I2h0cd9tdvc

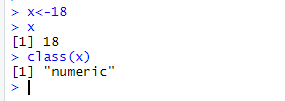
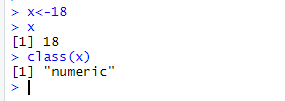
dir() function lists all the files in a directory.



ls() function is used to list the names of all the objects that are present in the working directory.

Description: https://lh4.googleusercontent.com/3SOwESpOP8H9GgrvLe1lGnZAAMeVoysn3g7Ah6NKVwXqmV2zMM9DuRRm_qeKWTghbAW2R261aNX8BNueBRm0DIdSmZkGZzuGnbkkO1OPHlp1wkrkU2jtWW3N1pthmRi376n4C3V3jmkQnJoaxGxaKkg

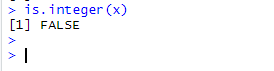
The class(x) function is used to return the values of the class attribute of an R object.



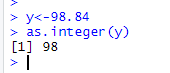
is.character() function is used to check if type of vector x is character.

Description: https://lh6.googleusercontent.com/Oz266UUedCPFqzBNFGwcgTtwIqi2IMeoiWczEHs3b5HdUgHPq_8uZ2zwpNQsru5AFamtcNunsjJAkOMX9oC7RpsCJ2D00O1GgzxpKtWSKnw9nHMtAjKffTCIAvz1OEm7QE6mVDzGpw6-d5sUQZ6xLUg

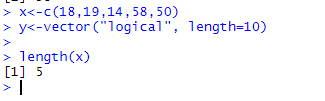
is.integer() function tests if a given number is an integer.



as.integer() function in is used to convert a character object to integer object.



c() function is used to create vector or list of items.

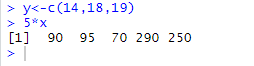


vector() is used to produces a vector of the given length and mode.

Description: https://lh5.googleusercontent.com/n2lUD8jRLb-mczQ6RNVgKkgQh0HyWLIK5Yc-O1cI1O5MnaNu0mkI2Ggwf4jXVUif4JLjN_ADThXzONdm_QPlYyEkYmR-Sdjn1X6f7ha4gKjec1NbZhq1s21wATuQo7_xzusXBsQAkzNE_XNzCjSIz5s

Vector operations: Various arithmetic operations can be performed member-wise.

Multiplication by a scalar.



Addition of two vectors.

Description: https://lh3.googleusercontent.com/9MiMvyHZmItIVR7Dd4b00T5LX635EFY6Q4lGLbGVvOhNxyv-CcL2-5jqK4IKC6Sr2nR5cVR2Nz02Y3Jw0INmBSmzNe4Qs-pCQQ0yQ_u88guGD7zOTmWP3AWi_75eBTkdNHuofy0yj1XGu_dnl5m2ElE

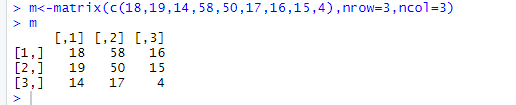
Multiplication of two vectors

Description: https://lh3.googleusercontent.com/L7fxzUIufjqiUpySCWXaK7rziB35sN3X7nWcO39y2EhbcVrzTtPxN4C-WpYhZ-nH0pZQKAjgdQ5UoejdQEGyDHaASz542hrLHYrgnhKz2LlMsIvE01qfq7hGxO7nLZyOwubsV5YHjmfbRW8nJnlZy7E

x to the power y

Description: https://lh5.googleusercontent.com/TquZrBfLJqSatWUHLRbFLB3TwqZv-gEZLENyfxyPuLGDOaaqEf9bUyGwVYPimxNdPOZ9Aa_bJv8ArG2HI5f_m91gAx1JRBGPov7fAG1VN7QhClVRK0qo-pw0uAXxQcomtidMBj93x1ClADIUn62RUw0

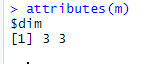
matrix() function used to create matrix of 2-D array having elements of same class.



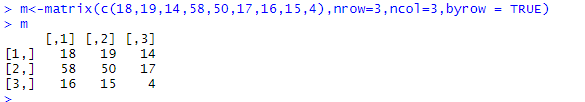
dim() dimensions of matrix m

Description: https://lh4.googleusercontent.com/xic9AJ9CE34p1VXGkD0aZD9SX0Sm8STn5wzId5zSR7blpIdiYtPm5SILloRyQVBJgpC5-tA2rANGJBjZwo4OzcmsYGvz2Bq6MDVLnw5wPeDabzmGqv1lYQRfs2k7T4sHa-1_QcOoREq7Tz_bfOtLCx0

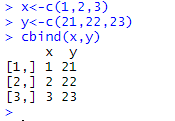
attributes() attributes of matrix m



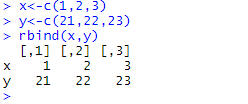
Creating another matrix:



Use the cbind() function to add additional columns in a Matrix.



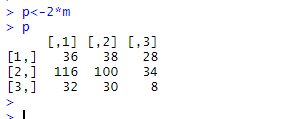
Use the rbind() function to add additional rows in a Matrix.



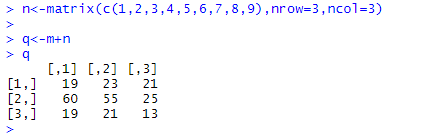
rbind() or cbind() function to combine two or more matrices together:

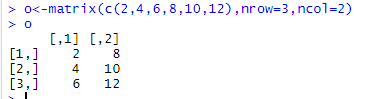
Matrix operations/functions:

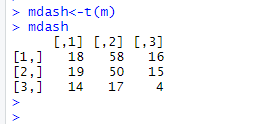
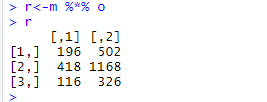
Multiplication by a scalar.



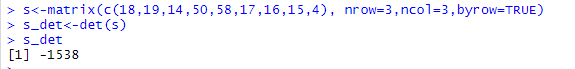
Addition, subtraction and multiplication of two matrices.



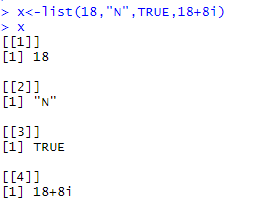




det() function is used to calculate the determinant of the specified matrix.

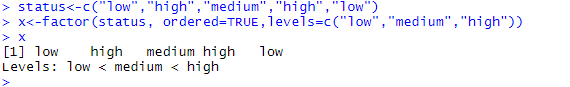
t() functions is used to transpose the rows and columns.

list() is used to create a list.



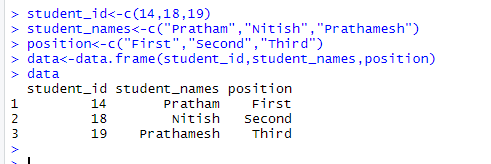
Factors are used to categorize data.

To create a factor, use the factor() function and add a vector as argument.



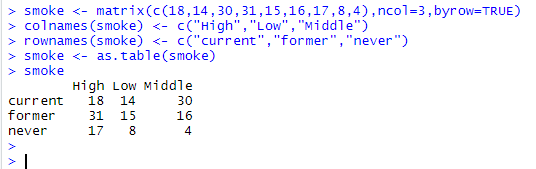
Data Frames are data displayed in a format as a table. Data Frames can have different types of data inside it. While the first column can be numeric, the second and third can be character or logical. However, each column should have the same type of data.

data.frame() is used to create a data frame.

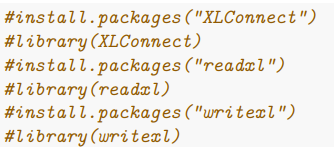


Description: https://lh5.googleusercontent.com/uI4eol-4UyTCSeZZPyS8Zu5FIfY21dLyG4LwCAJS5mMF42sj5uA_oIpawTmt42OKVoOC8BkXzcxLospUq1tGEq4rW7SbzK2uQZgElaGux6zPidt82dQOSxm88yakfTbJfbrHGYDsvMhFM4D837ssY04Description: https://lh3.googleusercontent.com/aaFan1Y-u1rWd14jLgOUVqNvHrS9PXInvJ0bxNRW3sr5fYzjuxv5X0Z17OECcU-lypIRvaIfk_cWEst6HdR1vTD92vSzkY9Bc4teJZKsJMB4ye6bMdIN-uNBeF0FmC92AJlQ3DCMlTR_aX1NP-3Z8cUDescription: https://lh5.googleusercontent.com/zhy_UFeEQNmJ4WViyHtfko-F881B89ScMN9J76CdJ46HnWgeqCx0J21_fhkss9zf0D52-jr9275kxM2IVqvgTN3toobJFQUxn8s2APpIpOJIK1AHPw5OaWAEq1lJAJkO-oww6z15yl9dUH0HSFbn_ocDescription: https://lh3.googleusercontent.com/W_KFHzyxsx7wPKCMyHOEpz4dFxV9jdJW_epQfjgl2isUpUbqQzqihZFRnZbEiiXCFun4VMruq4Br8fb5YMAPYgIA1C0NZsE5TZvTeczUA_p9UWHKn6XluIl-Mqadu18SBWoz3DTQvrt4OKPa-UWHisw

as.table() function is used to convert an object into a table.

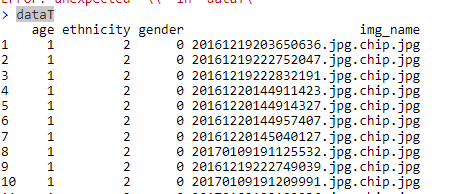


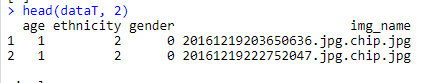
Reading and writing data from csv:

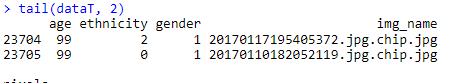


dataT <- read.table("age\_gender.csv", sep =",", header = T)

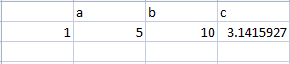
dataT



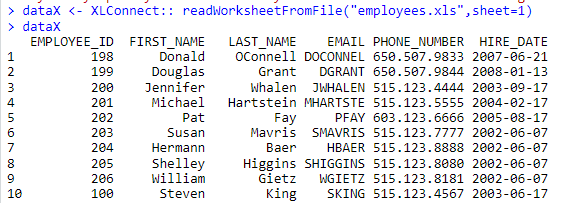
Description: https://lh6.googleusercontent.com/qnYHTWThMwG2pWLVA4G_wem_CPEkN3ZqRn5AUQHX-7wLwxdUp2Qsi14J0FfSy9pI6NBSZ4eyFuWaXa5YTFaJvLqqleyF5GuAylVwY5W806wh8BFcjInC2N0fW2apKgJgPXH8UbnmsuJuSGdf-PYoXtE



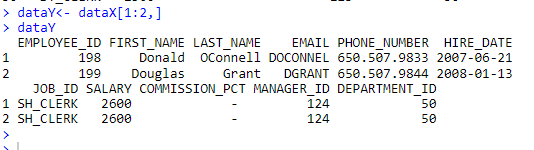
write.csv() is used to write to csv file

Description: https://lh4.googleusercontent.com/nusjZidi5GuX84gdn9RnfWcgRgQQbzjBahSxECB9NIp09J1wIWySQ15Ntm7b75Qo912-Qk1w4-h8wScxdNKWHggiduzpyBL8Goh10lMiBH_7AFbHg9yw7EnkBj_jorkP3QmX4M1U_s-4UlS1lExw6S-4YbBetub1

readWorksheetFromFile() is used to reads data from worksheets in an Excel file.



Slicing of the Excel file:



CONCLUSION: I have gained knowledge regarding the basics of R and data acquisition.

**PRACTICAL 6**

AIM: Implementation of Data preprocessing techniques like, Naming and Renaming variables, adding a new variable, Dealing with missing data, Dealing with categorical data, Data reduction using subsetting.

THEORY:

Preprocessing in Data Mining:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Steps Involved in Data Preprocessing:

Data Cleaning: The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

Missing Data: This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

Ignore the tuples: This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values: There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

Noisy Data: Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

Binning Method: This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

Regression:Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

Clustering: This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

Data Transformation: This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

Normalization: It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

Attribute Selection: In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

Discretization: This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

Concept Hierarchy Generation: Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

Data Reduction: Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs. The various steps to data reduction are:

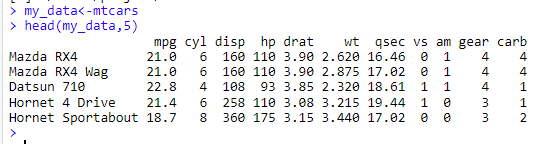
Data Cube Aggregation: Aggregation operation is applied to data for the construction of the data cube.

Attribute Subset Selection: The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.

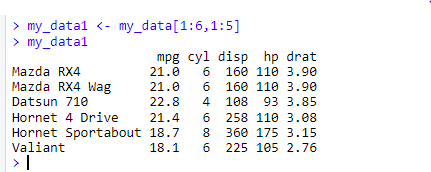
Numerosity Reduction: This enable to store the model of data instead of whole data, for example: Regression Models.

Dimensionality Reduction: This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Component Analysis).

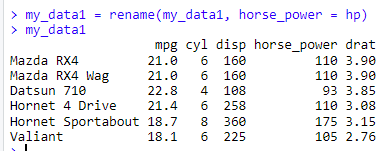
1. Loading mtcars dataset and printing first 5 records:



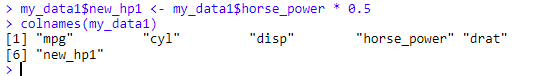
1. Displaying first six records with only five columns of mtcars dataset:

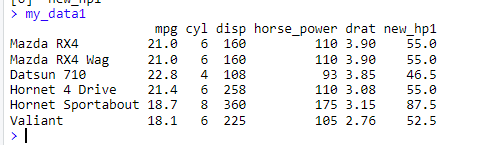


1. Renaming hp column name to horse\_power using dplyr package:

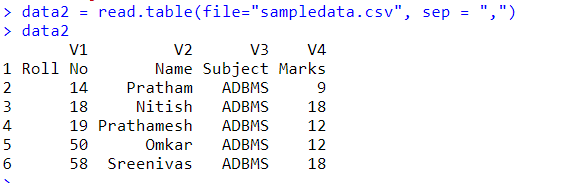


1. Adding new column names new\_hp to mtcars dataset:

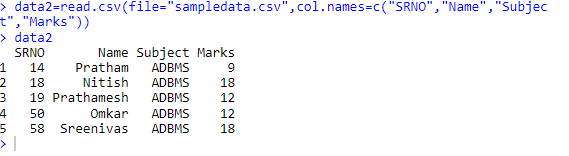




1. Reading data from csv file using read.table()



1. Reading data from csv file using read.csv() and rename default column names



1. Displaying operations with NA

Description: https://lh5.googleusercontent.com/aLAHnFh7mJDee2G1-fF3BtpjrOIohhKTodCAI2ciOnqxMEjm61Ok1_HgFEcd76Zv38gXWpkcOsKvzHiGXCOS91PcSmhl8idaFt4FOE_ghNUfPlLFCV7Rd3PUyVXQ35AnQFo0QJgeIsdySoTelG8lGjU

1. Creating a vector with NA values and calculating its median

Description: https://lh5.googleusercontent.com/a6xG94WmVlHc9C57zPIEjC-B8_uvaNnSJM8-VInfQ4qR0_Txw5styf4q63WMEE6cE9J2PvG9UppSaFJF2aRNO3u5fvsYCAwWDRYA3TTBZsFG9ij0Vqu5AY9tnDfcs61UU7dZXhS8bnZ8DORl8FqHFSo

1. Checking whether NA values are present or not in the vector

Description: https://lh6.googleusercontent.com/YhcfUVY0XqKKLNsCS0ZPLTNAXpQon9g8QC_Oi0hIxheOiHFytrIBrLlMa3y1Mf2ceOte36LfYOg7L_khIGZlRNwhBVLAO7UGCDhKiYZhZBrSRxxzxWiogxBtOfGBAN6hsBqWYXGYqoOs4E-35cOsgy4

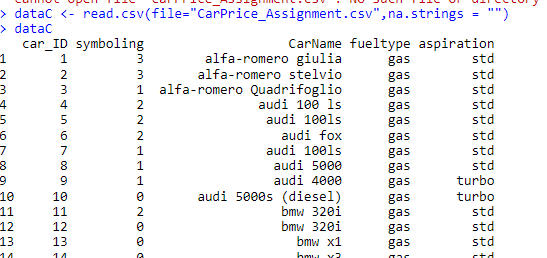
10.Removing NA values by logical indexing and display them

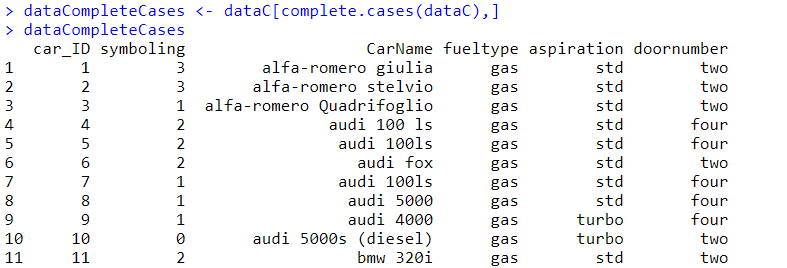
Description: https://lh6.googleusercontent.com/O_bYVf3GZ8NrBxb9PzwflvX9Fn0GtVsAe41JfDhKXtHknNBF5ZFMfb9VITtKTS5EZD0AE6MfsLhMzjyUSl_RVOXusYuAq3URySXbRasoDKpLArRz8KHtG0U88NgQ-MDXZgws14lJb97hQoGQQh722Wc

11.Subsetting with complete cases (values that are not NA)

Description: https://lh5.googleusercontent.com/62ypVHMKoT4G4WJg7GwX_vkhopTdImezUgFgyf6FWFuCAjAGE9KOT9-w06_qXtC-VRxrrEn33aYv7fyf5u206TbmiYd1gu4X7L6zyTnNnJWgaD7UhpfYtLJvKpg6YhjuwgvCzjuyKh2XfNejn0TcHp4

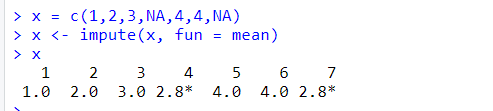
12. Subsetting data frame using complete.cases



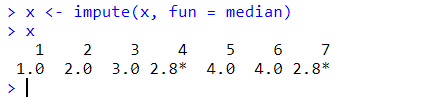


13. Mean imputation

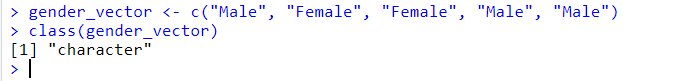
library(Hmisc)

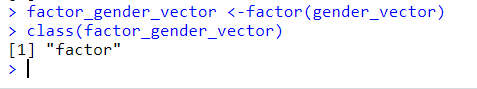


14. Median imputation

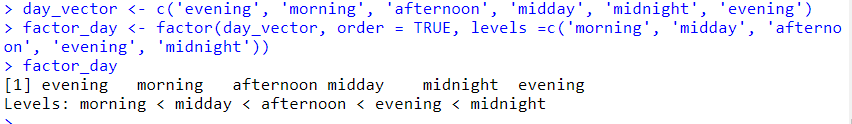


15. Converting gender vector(consisting of categorical data) to a factor

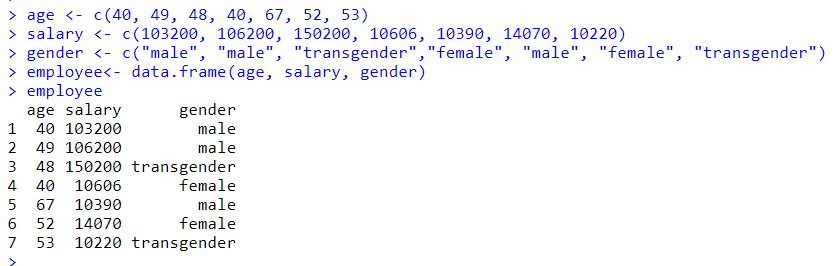




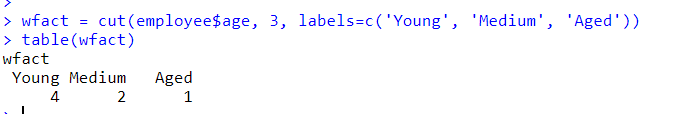
16. Creating ordinal categorical vector and converting vector to a factor with ordered level



17. Creating dataframe from the vectors



18. Creating a factor with labels



**PRACTICAL 7**

AIM: Implementation and analysis of Linear regression through graphical methods.

THEORY:

Linear Regression:

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables.

1. One of these variables is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.
2. In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1.
3. Mathematically a linear relationship represents a straight line when plotted as a graph.
4. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

The general mathematical equation for a linear regression is y = ax + b

Following is the description of the parameters used −

y is the response variable.

x is the predictor variable.

a and b are constants which are called the coefficients.

Steps to Establish a Regression: A simple example of regression is predicting weight of a person when his height is known. To do this we need to have the relationship between height and weight of a person.

The steps to create the relationship is −

a) Carry out the experiment of gathering a sample of observed values of height and corresponding weight.

b) Create a relationship model using the lm() functions in R.

c) Find the coefficients from the model created and create the mathematical equation using these

d) Get a summary of the relationship model to know the average error in prediction. Also called residuals.

e) To predict the weight of new persons, use the predict() function in R

Error Calculation:

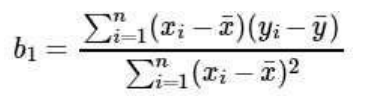
Description: https://lh6.googleusercontent.com/xZBDf5MAznq3AshyT9HFI526N6TKYcRjRKHDYEod-L0aEGyu5FA8lzI-znAPChyzIAxpOPfNF7qbVtIh6z-SI3-NqX7IJhlmYkgoG9W0ViBD8kRGQJvPrrryG0UqBaPZvFSHri1ySIip0Py9L2A4ioQba4cWBFUB

If we don’t square the error, then positive and negative point will cancel out each other.

Intercept Calculation:

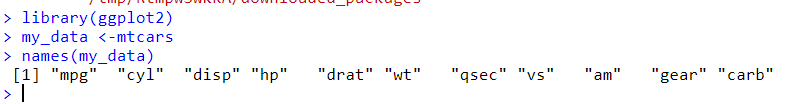
Description: https://lh4.googleusercontent.com/bURR24q8mHVwgQca9y5uYQTSfkjiCwXg4Z5W--fEsnpS6yEGVVwA6xPjBg2KQQfOP7GP23KGP4nUWobS4qJ05Zm7VKvfnLYvQ7_J4lUL_IxVegtnC7L32v5kFItzrfu9AHIGrQKHUVSU0yT0XCMusOAwdcO8jZLI

Co-efficient Formula:



1.Loading "ggplot2" library, "mtcars" dataset and displaying the attributes names and dimension for that dataset.

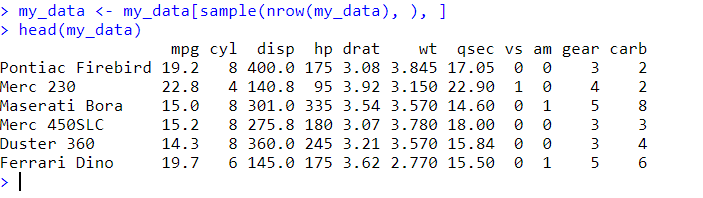
#install package tiddyverse



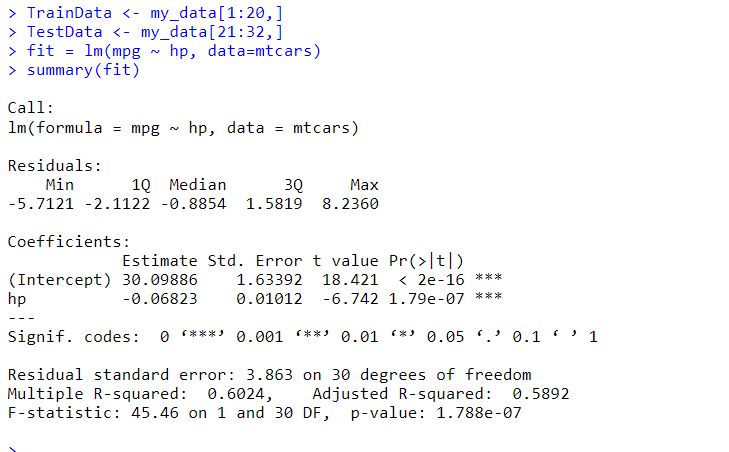
2. Display dimensions of the dataset

Description: https://lh3.googleusercontent.com/5JG6te2Wm3QdhLJIqI-aYCS20C62QoLMkrET9oeFAB8VZni_xyaKfyRDDhYFXqv_hBw2eG_UV6h5x6J1M_nyfZC7O-YzMAZW9b5LxWvMADBjzv4WssiZvHdfRQn5lM3kryqrw6uswJBTgA24CPdpZx4

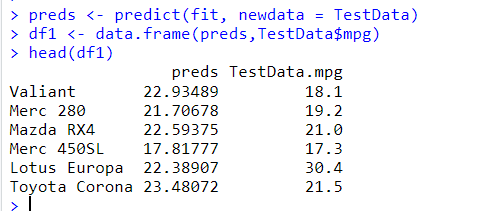
3. Displaying first 5 sample rows of dataset.



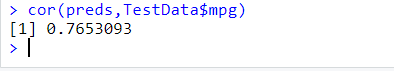
4. Splitting the dataset rows in 62:38 ratio where 62%=20 rows for training the model and remaining 38%=11 rows for testing the model and Fitting the dependent(mpg) and independent(hp) attributes of mtcars data into the linear model and displaying the model summary.



5. Predicting or Testing the trained model by passing the test data to the model and displaying the predicted mpg along with the actual mpg using dataframe.

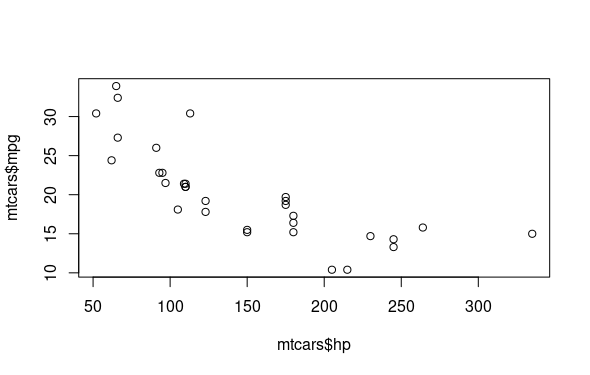


6. Getting the correlation accuracy between predicted data and actual data.



7. Scatter plot graph of hp and mpg columns of mtcars dataset

plot(mtcars$hp, mtcars$mpg)



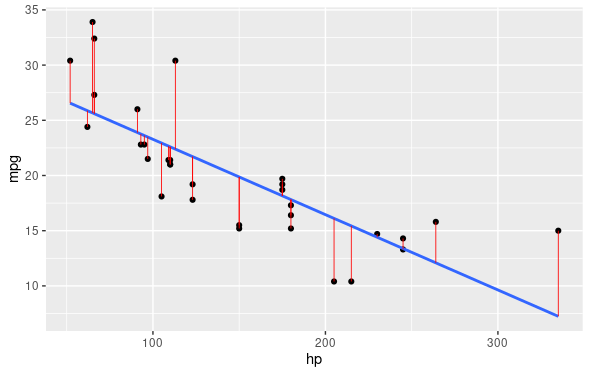
8. Plotting the predicted regression line on actual regression data point and also displaying the residuals between actual and predicted values.

ggplot(fit, aes(hp, mpg)) +

geom\_point() +

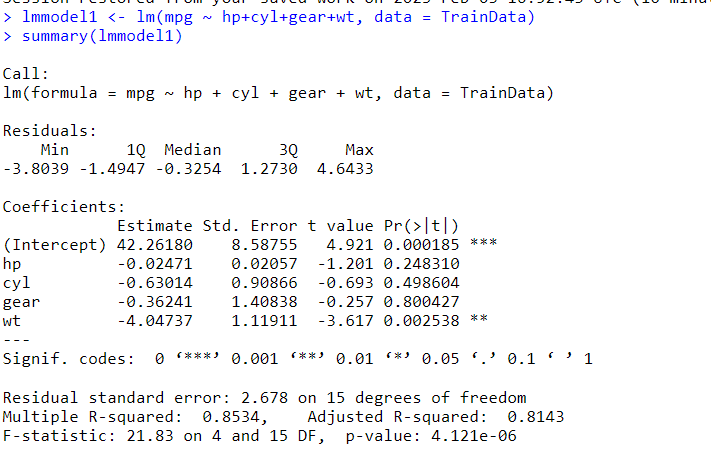
stat\_smooth(method = lm, se = FALSE) +

geom\_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)

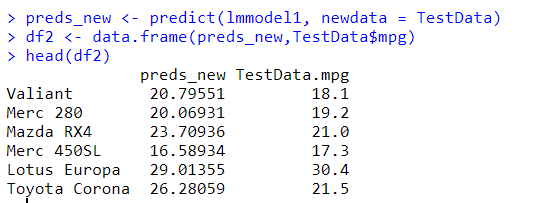


Multivariate Linear Regression:

1.Fitting the dependent(mpg) and other independent(hp, cyl, gear,wt) attributes of mtcars data into the linear model and displaying the model summary



2. Predicting or Testing the trained model by passing the test data to the model and displaying the predicted mpg along with the actual mpg using dataframe.

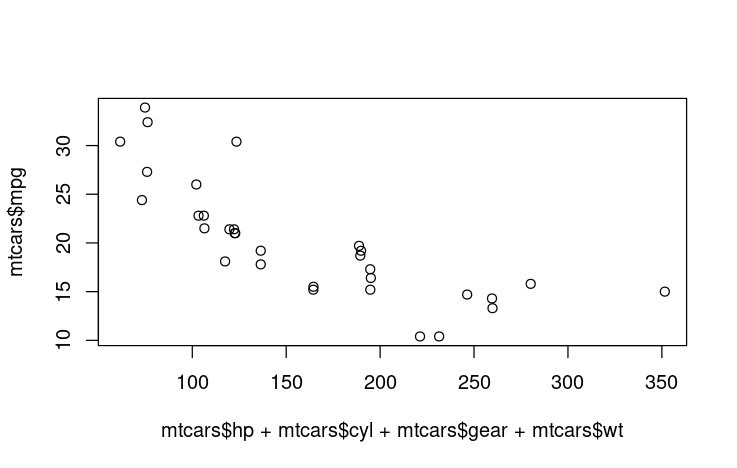


3. Getting the correlation accuracy between predicted data and actual data.

Description: https://lh4.googleusercontent.com/b3epnLaNYTJS8W-CUhgMXdWT0jpqtfMpvtDOvq_47a1fTaQWlbd9oOdGdEHotNZ8EhXuc_DL0OxSLp8iEGnO400rgOwnBrFWbVA3BQtkQTo9vLhSMBO7eAoUhJqXElGpOYhaVyfySYAw8zWX4x3DqCw

4. Scatter plot graph of (hp,cyl,gear,wt) and mpg columns of mtcars dataset

plot(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mtcars$mpg)



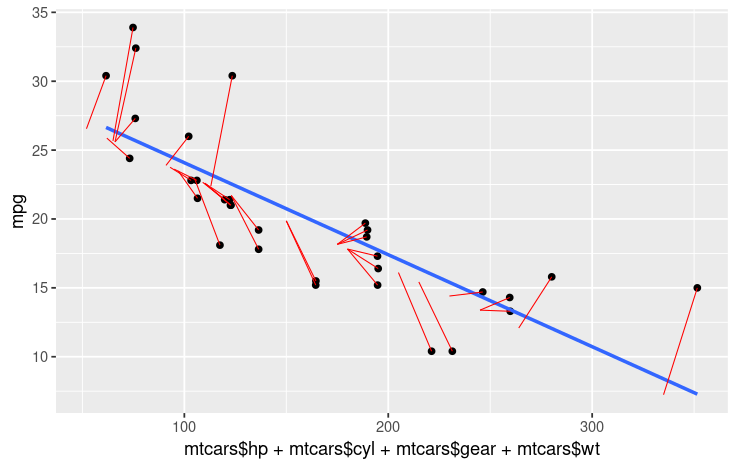
5. Plotting the predicted regression line on actual regression data point and also displaying the residuals between actual and predicted values.

ggplot(fit, aes(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mpg)) +

geom\_point() +

stat\_smooth(method = lm, se = FALSE) +

geom\_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)



CONCLUSION:

I have gained knowledge regarding the implementation and analysis of Linear regression through graphical methods.

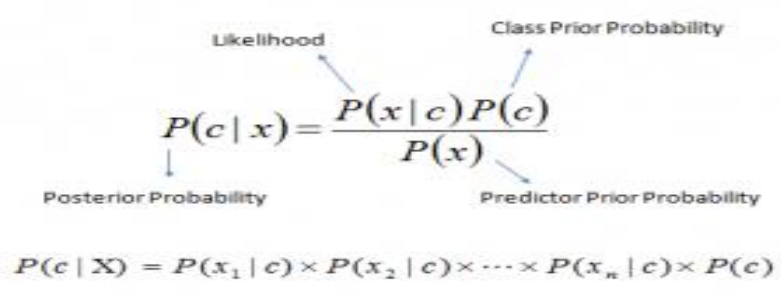
**PRACTICAL 8**

AIM: Implementation and analysis of Classification algorithms like Naive Bayesian,K-Nearest Neighbor.

THEORY:

Naïve Bayes Algorithm:

1. It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors.
2. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
3. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.
4. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.
5. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below



Above,

a) P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).

b) P(c) is the prior probability of class.

c) P(x|c) is the likelihood which is the probability of predictor given class. d) P(x) is the prior probability of predictor.

Working:

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Pros:

a)  It is easy and fast to predict class of test data set. It also perform well in multi class prediction

b) When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.

c) It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

a) If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.

b) On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.

c) Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications:

a) Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

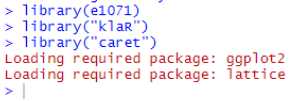
b) Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.

c) Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

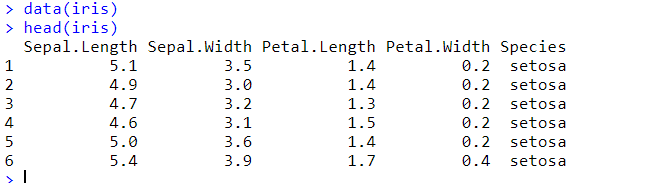
d) Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

Execution:

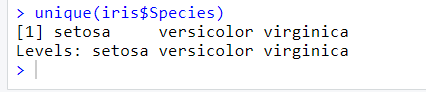
1.Loading the libraries viz “e1071” , “klaR” , “caret” , “ggplot2”.



2. Loading and displaying the structure of iris dataset



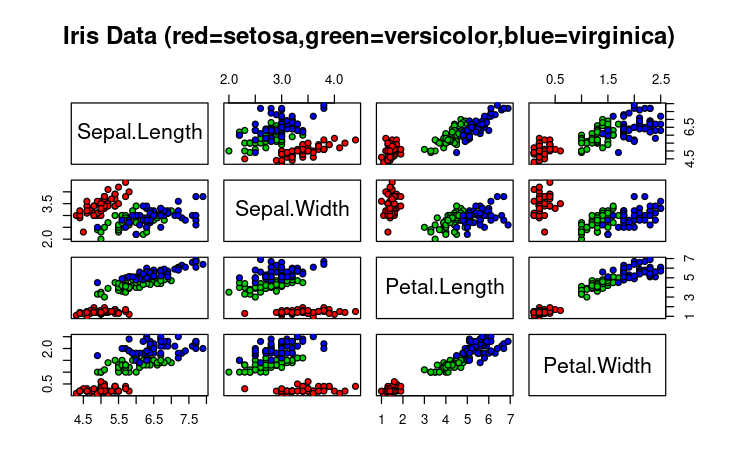
3. Displaying unique values from Species attribute



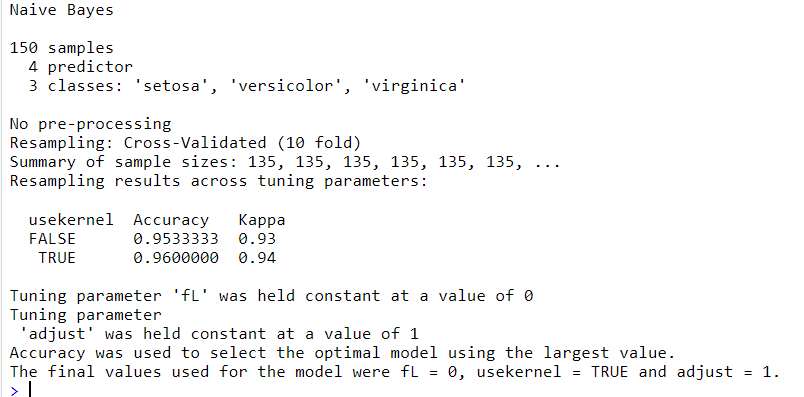
4. Plotting the relationship graph of all attributes of iris dataset

pairs(iris[1:4], main="Iris Data (red=setosa,green=versicolor,blue=virginica)",

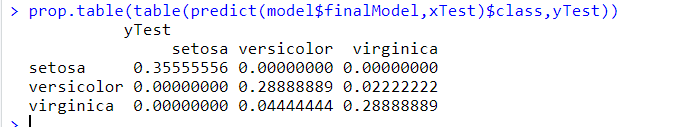
pch=21, bg=c("red","green3","blue")[unclass(iris$Species)])



5. Splitting the dataset into 70:30 where 70 for training sets and 30 for testing sets and Building the model.



6. Predicting the data and displaying through frequency table



K- Nearest neighbor Algorithm:

1. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
2. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
3. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
4. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
5. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
6. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
7. Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Working:

The K-NN working can be explained on the basis of the below algorithm:

a) Step-1: Select the number K of the neighbors

b) Step-2: Calculate the Euclidean distance of K number of neighbors

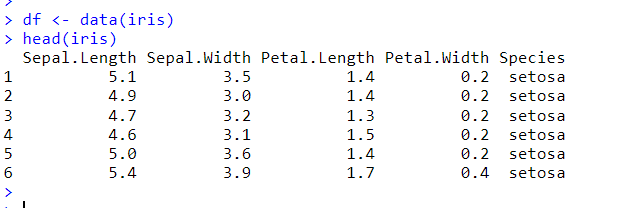
c) Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

d) Step-4: Among these k neighbors, count the number of the data points in each category.

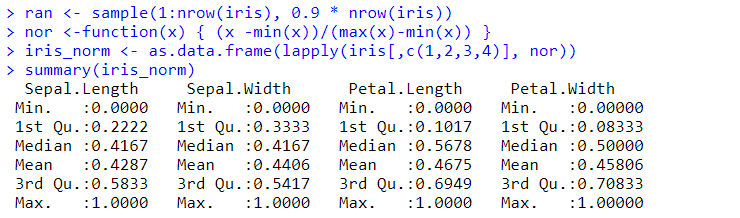
e) Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

f) Step-6: Our model is ready.

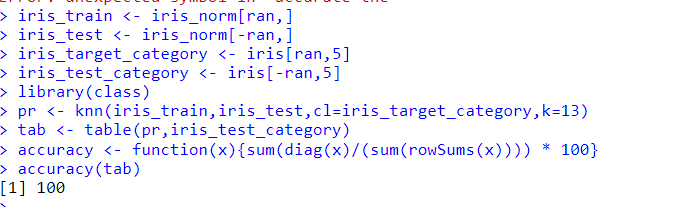
1.Loading and displaying the structure of iris dataset.



2. Generating random number that is 90% of total number of rows in data, Creating normalization function and applying this function to first 4 column of dataset as they are predictors and Displaying the summary of normalized dataset which contains Min, Max, all Quartiles and mean.



3. Creating training and testing sets using the sample sets of 90% of normalized data for training and 10% for testing. Extracting 5th column of test dataset to measure the accuracy Loading the package class and creating the knn function.Creating confusion matrix and Displaying the accuracy of our knn function using confusion matrix by dividing the correct prediction by total number of prediction.



CONCLUSION:

I have gained knowledge regarding the classification algorithms like Naïve Bayes and K Nearest neighbor algorithms.

**PRACTICAL 9**

AIM: Implementation and analysis of Apriori Algorithm using Market Basket Analysis.

THEORY:

Apriori  Algorithm:

1. The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.
2. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data.
3. Apriori is designed to operate on database containing transactions (for example, collections of items bought by customers, or details of a website frequentation).

Key Concepts:

1. Frequent Itemsets: All the sets which contain the item with the minimum support (denoted by Li for ith itemset).
2. Apriori Property: Any subset of frequent itemset must be frequent.
3. Join Operation: To find Lk, a set of candidate k-itemsets is generated by joining Lk-1 with itself.

Steps to perform Apriori Algorithm:

Step 1 - Scan the transaction data base to get the support of S each 1-itemset,

compare S with min\_sup and get a support of 1-itemset, L1

Step 2 - Use Lk-1 join Lk-1 to generate a set of candidate k-itemsets. And use

Apriori property to prune the unfrequented k-itemsets from this set.

Step 3 - Scan the transaction database to get the support S of each candidate k-itemset in the find set, compare S with min\_sup and get a set of frequent k-itemsets Lk

Step 4 - If the candidate set=NULL then go to step 5 else go to step2 .

Step 5 - For each frequent itemset 1, generate all nonempty subsets of 1

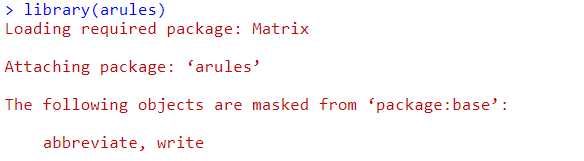
Step 6 - For every non empty subset s of 1, output the rule “s=>(1-s)” if confidence C of the rule “s=>(1-s)” (=support s of 1/support S of s) min\_conf.

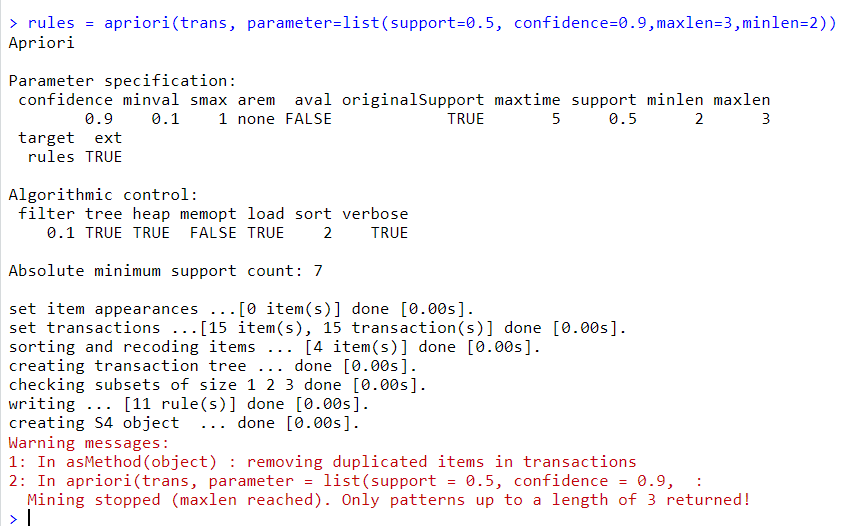
Execution:

1.Reading the csv file ‘data\_apriori.csv’ using read.csv() and Creating the transaction by splitting the products attribute of the dataset and grouping them according to their customer id.

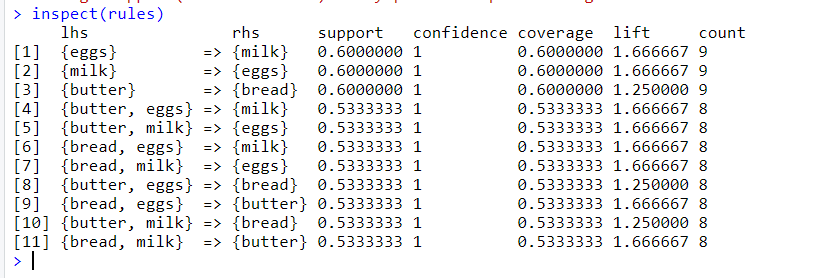


2. Loading the ‘arules’ library and applying the apriori function with minimum support of 0.5 = 50%, confidence of 0.9 = 90% and limiting the element in a rule to 3





3. Summarizing the transaction rules by apriori function



CONCLUSION:

I have gained knowledge regarding the implementation of Apriori algorithm in R.

**PRACTICAL 10**

AIM: Implementation and analysis of Clustering algorithms like K-Means, Agglomerative Assessment.

THEORY:

K-MEANS CLUSTERING:

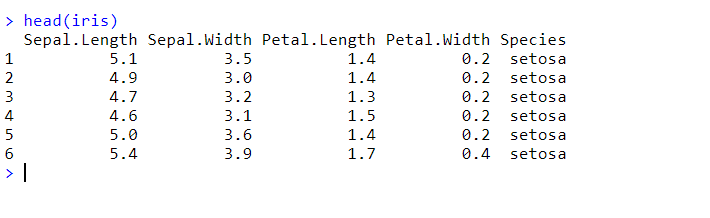
* K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabelled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
* It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.
* It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
* The algorithm takes the unlabelled dataset as input, divides the dataset into k number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
* The k-means clustering algorithm mainly performs two tasks:
  1. Determines the best value for K center points or centroids by an iterative process.
  2. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
* Hence each cluster has data points with some commonalities, and it is away from other clusters.
* The working of the K-Means algorithm is explained in the below steps:
  + Step-1: Select the number K to decide the number of clusters.
  + Step-2: Select random K points or centroids. (It can be other from the input dataset).
  + Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
  + Step-4: Calculate the variance and place a new centroid of each cluster.
  + Step-5: Repeat the third steps, which means assign each datapoint to the new closest centroid of each cluster.
  + Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
  + Step-7: The model is ready.

AGGLOMERATIVE CLUSTERING:

* The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It’s also known as AGNES (Agglomerative Nesting).
* The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram.
* Agglomerative clustering works in a “bottom-up” manner. That is, each object is initially considered as a single-element cluster (leaf).
* At each step of the algorithm, the two clusters that are the most similar are combined into a new bigger cluster (nodes).
* This procedure is iterated until all points are member of just one single big cluster (root)
* Steps to agglomerative hierarchical clustering:
* We’ll follow the steps below to perform agglomerative hierarchical clustering using R software:
  + Preparing the data
  + Computing (dis)similarity information between every pair of objects in the data set.
  + Using linkage function to group objects into hierarchical cluster tree, based on the distance information generated at step 1. Objects/clusters that are in close proximity are linked together using the linkage function.
  + Determining where to cut the hierarchical tree into clusters. This creates a partition of the data.

K-Means Clustering:

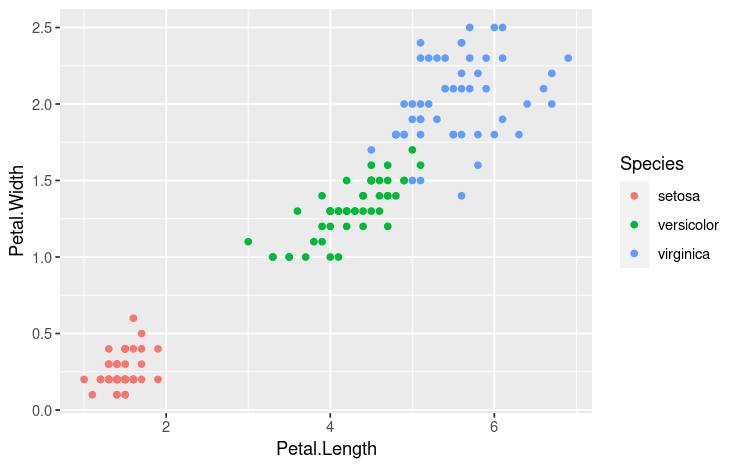
1.Load the iris dataset and display the first eight records.



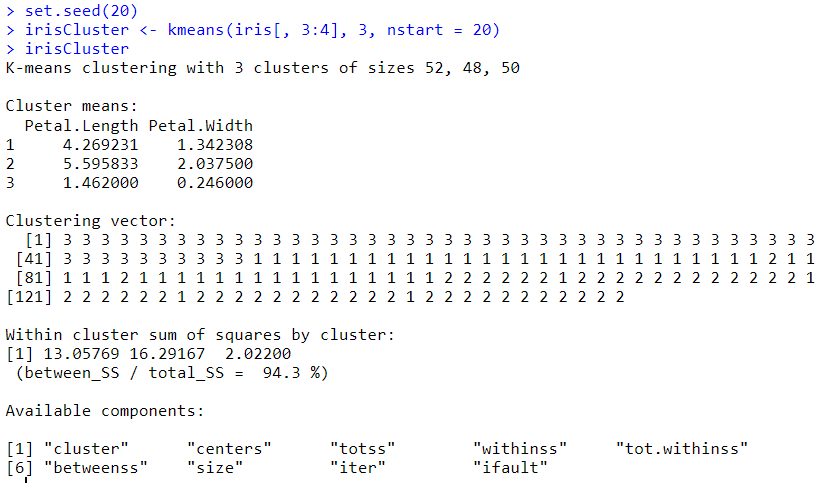
2. Plot the graph.

library(ggplot2)

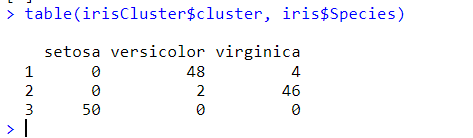
ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom\_point()



3. Create 3 clusters for Petal.Length and Petal.Width columns of iris dataset using k-means clustering algorithm.



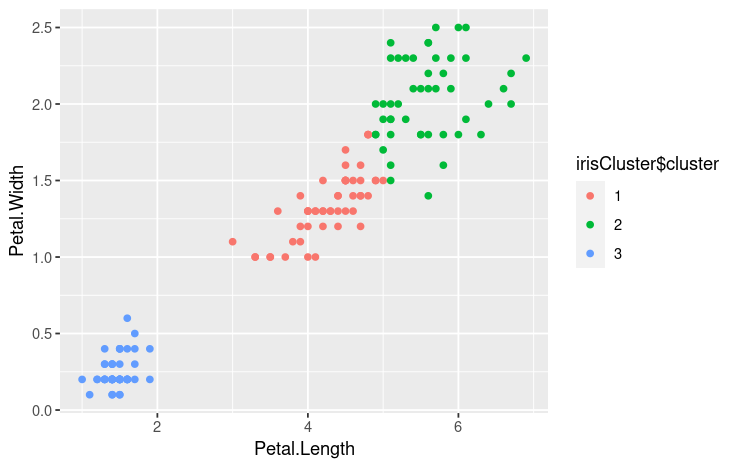
4. Display the frequency of irisCluster$cluster and iris$Species.



5. Plot the graph.

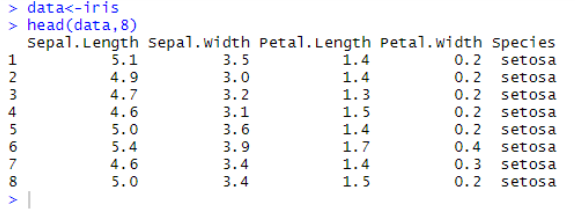
irisCluster$cluster <- as.factor(irisCluster$cluster)

ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point()

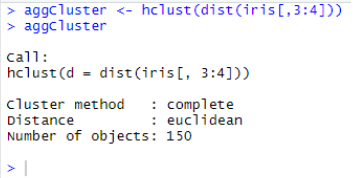


Agglomerative Clustering algorithm:

1.Load the iris dataset and display the first eight records.

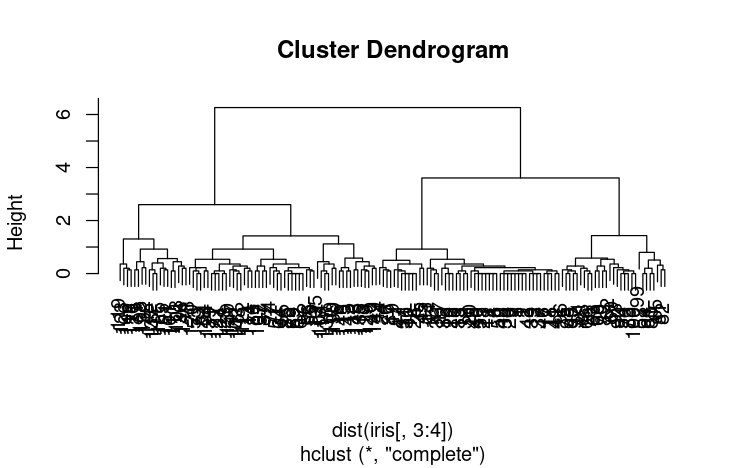


2. Performing a hierarchical cluster analysis :

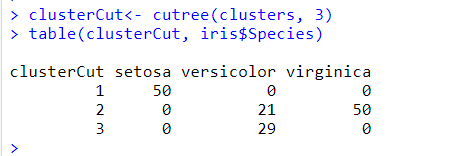


3. Plot the graph:

plot(clusters)

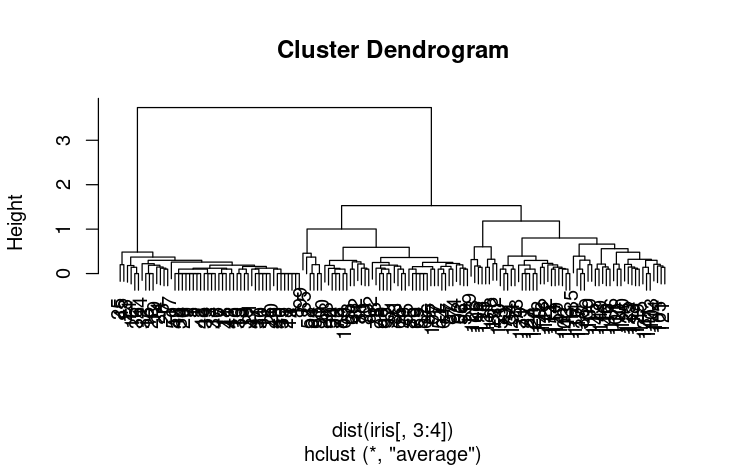


4. Frequency Table of cluster:



clusters <- hclust(dist(iris[,3:4]), method = 'average')

plot(clusters)

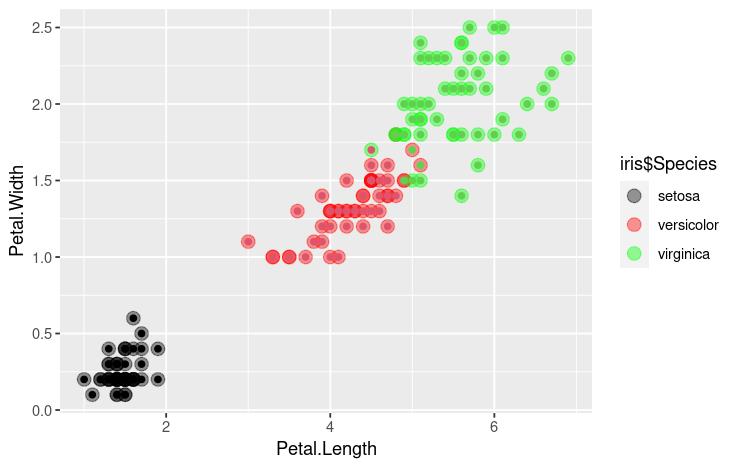


5. Creating model and plotting the graph:

clusterCut <- cutree(clusters, 3)

table(clusterCut,iris$Species)

ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) + geom\_point(alpha = 0.4,   size = 3.5) + geom\_point(col = clusterCut) + scale\_color\_manual(values = c('black','red', 'green'))



CONCLUSION:

I have gained knowledge regarding the implementation of k-means clustering and agglomerative clustering algorithms.