

**TRAINING REPORT**

**OF**

**SIX MONTHS INDUSTRIAL TRAINING, UNDERTAKEN**

**AT**

**“CHITKARA UNIVERSITY”**

**IN**

**“CURIN DEPARTMENT”**

**ON**

**“DATA ANALYSIS”**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE**

**OF**

**Masters of Computer Application**

**Under the Guidance of: Submitted By:**

**Name: Mr. Rajesh Kaushal Name: Ashish Soni**

**Designation: Associate Professor College Roll No.: 1510981019**

**Department: Computer Applications University Roll No.: 1510981019**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY,**

**VILL. JHANSLA, TEHSIL RAJPURA,**

**DISTT. PATIALA 140401**

Date :23rd May 2017

**Certificate of Training**

This is to certify that Mr. Ashish Soni student of MCA of Chitkara Institute of Engineering and Technology, Rajpura, has successfully completed six months industrial training from January to May 2017. During this training, he worked on Data Analytics in CURIN department under the guidance of Mr. Rajesh Kumar Kaushal. His overall performance during the training period was excellent.

(Signature)

Rajesh Kumar Kaushal

Associate Professor

Department of Computer Applications

(Signature)

Dr. Surya Narayan Panda

Director, Research

CURIN

**Acknowledgement**

I have put in great efforts to complete this project. However, it would not have been possible without the kind support and help of many individuals who assisted me in this project. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. S.N. Panda, Mr. Rajesh Kaushal** and **Mr. Jaspreet Singh** for their guidance and continuous supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents and members of **CURIN Department** for their kind co-operation and encouragement which helped me in completion of this project.

I would also like to express my special gratitude and thanks to every person for giving me such attention and time.

My thanks and appreciation also goes to the **Department of Computer Applications** in developing the project and people who have willingly helped me out with their abilities.

**Preface**

The project has been made by my own effort that I have learned in my internship period. This project is based on Data Analysis that includes analyzing a given dataset. Through the acquired knowledge about Hadoop and R, I have tried my best to present this project report properly. In preparing this project, I have taken help from many papers and online resources available on internet.

**Ashish Soni**

**CONTENTS**

|  |  |
| --- | --- |
| Contents | Page no |
| Certificate of training | 2 |
| Acknowledgement | 3 |
| Preface | 4 |
| Company Profile | 6 |
| Introduction to assigned job | 7-14 |
| Description of the job | 15-16 |
| Project | 17-44 |
| Industry application | 45 |
| Conclusion | 46 |
| References | 47 |

**COMPANY PROFILE:**

Chitkara University is recognized by University Grants Commission (UGC), a statutory body of the Government of India established for the coordination, determination and maintenance of standards of university education in India.

In the year 2010, Chitkara University was established by the Punjab State Legislature under, “The Chitkara University Act”. The University has, at the center of its philosophical core, the commitment to excellence in education. It is the strong academic heritage of the University, and such firmness on matters of principles that has seen it grow from strength to strength in such a short span of time.

CURIN (CHITKARA UNIVERSITY RESEARCH & INNOVATION NETWORK)

Department is dedicated towards research and innovation in various fields or trades that are a part of the University’s up to date curriculum. There are various research centres and institutes are the locus of research for collaborative groups of investigators pushing the frontiers of knowledge forward.

They are involved in progressive research while exploring new technologies in order to create highly efficient devices, smarter applications and using wireless sensors networks that improve the country’s infrastructure and safety — and contributing to society through many other discoveries and innovations.

The mission of CURIN is:

“To promote interdisciplinary and applied research beneficial to our University, Society, National and International Community and

To incorporate and inculcate inclusive research habits – with innovative streak – at all levels of students, employees and faculty.”

Key roles of the Centres include:

* Maintain an institutional reputation for innovation and responsive education and training delivery to their targeted industry.
* Translate industry research into best practices.
* Provide system coordination, coaching, and mentoring to assist in building seamless educational and work-related systems.

**Introduction to Assigned Job**

The project focuses on analysing data that has been stored on HDFS using language R. The dataset that has been used is a dataset about heart diseases and the various factors that can lead to an increase in the probability of having a heart disease.

The framework and language has been defined in this section and the in depth description to it is given in the following section.

HADOOP

Hadoop is an open source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Hadoop framework includes 4 modules:

1. Hadoop COMMON: These are the java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary java files and scripts required to start Hadoop.
2. Hadoop YARN: This is a framework for job scheduling and cluster resource management.
3. Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.
4. Hadoop MapReduce: This is YARN-based system for parallel processing of large datasets.



**MapReduce:**

Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

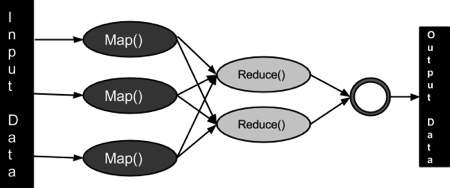
The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

**The Map Task:** This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).

**The Reduce Task:** This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.



**Hadoop Distributed File System:**

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data.

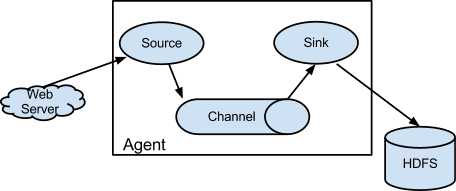
A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode.

HDFS provides a shell like any other file system and a list of commands are available to interact with the file system.

**Different tools of Hadoop:**

**Flume:**

Is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.



**Sqoop:**

Is a tool designed to transfer data between Hadoop and relational databases or mainframes. You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle or a mainframe into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.

Sqoop automates most of this process, relying on the database to describe the schema for the data to be imported. Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance.

**Kafka:**

Kafka is a distributed streaming platform. It is used for building real-time streaming data pipelines that reliably get data between systems or applications. Building real-time streaming applications that transform or react to the streams of data.

* Kafka is run as a cluster on one or more servers.
* The Kafka cluster stores streams of *records* in categories called *topics*.
* Each record consists of a key, a value, and a timestamp.

Kafka has four core API’s:

* Producer API
* Consumer API
* Streams API
* Connector API

**Pig:**

Is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig's language layer currently consists of a textual language called Pig Latin, which has the following key properties:

* **Ease of programming:** It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.
* **Optimization opportunities:** The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.
* **Extensibility:** Users can create their own functions to do special-purpose processing.

**Hive:**

Is a data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive.

* Pig and Hive have a similar goal- they are tools that ease the complexity of writing complex java MapReduce programs.

**Flink:**

Is an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications.

**2 types of datasets:**

* Unbounded: Infinite datasets that are appended to continuously.
* Bounded: Finite, unchanging datasets

**2 types of execution models:**

* Streaming: Processing that executes continuously as long as data is being produced
* Batch: Processing that is executed and runs to completeness in a finite amount of time, releasing computing resources when finished

**Spark:**

Is a fast and general engine for large-scale data processing.

* Is faster than Hadoop MapReduce.
* Easy to use.
* Generality: can combine SQL, streaming and complex analytics.
* Runs everywhere (Spark runs on Hadoop, Mesos, standalone, or in the cloud). It can access diverse data sources including HDFS, Cassandra, HBase, and S3.

**Storm:**

Is a free and open source distributed real-time computation system.

* Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing.
* Storm has many use cases: real-time analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Storm is fast: a benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate.
* Storm integrates with the queuing and database technologies you already use.
* A Storm topology consumes streams of data and processes those streams in arbitrarily complex ways, repartitioning the streams between each stage of the computation however needed.

**Beam:**

Beam provides an advanced unified programming model, allowing you to implement batch and streaming data processing jobs that can run on any execution engine.

Apache Beam is:

**UNIFIED:** Use a single programming model for both batch and streaming use cases.

**PORTABLE:** Execute pipelines on multiple execution environments, including Apache Apex, Apache Flink, Apache Spark, and Google Cloud Dataflow.

**EXTENSIBLE:** Write and share new SDKs, IO connectors, and transformation libraries.

**Apex:**

Enterprise-grade unified stream and batch processing engine.

* Apex is a Hadoop YARN native platform that unifies stream and batch processing. It processes big data in-motion in a way that is highly scalable, highly performant, fault tolerant, stateful, secure, distributed, and easily operable

.

* Enables developers to write or re-use generic Java code.
* **Modular:** The Apex platform comes with Malhar, a library of operators (units of functionality) that can be leveraged to quickly create non-trivial applications. Includes many connectors for messaging systems, databases, files etc.

**Samza:**

Is a distributed stream processing framework. It uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management.

* **Simple API:** Unlike most low-level messaging system APIs, Samza provides a very simple callback-based “process message” API comparable to MapReduce.
* **Managed state:** Samza manages snapshotting and restoration of a stream processor’s state. When the processor is restarted, Samza restores its state to a consistent snapshot. Samza is built to handle large amounts of state (many gigabytes per partition).
* **Fault tolerance:** Whenever a machine in the cluster fails, Samza works with YARN to transparently migrate your tasks to another machine.
* **Durability:** Samza uses Kafka to guarantee that messages are processed in the order they were written to a partition and that no messages are ever lost.
* **Scalability:** Samza is partitioned and distributed at every level. Kafka provides ordered, partitioned, replayable, fault-tolerant streams. YARN provides a distributed environment for Samza containers to run in.
* **Pluggable:** Though Samza works out of the box with Kafka and YARN, Samza provides a pluggable API that lets you run Samza with other messaging systems and execution environments.
* **Processor isolation:** Samza works with Apache YARN, which supports Hadoop’s security model, and resource isolation through Linux CGroups.

**Introduction to R:**

R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team, of which Chambers is a member.

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing.

The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data miners, and studies of scholarly literature databases show that R's popularity has increased substantially in recent years.

R is a GNU package. The source code for the R software environment is written primarily in C, Fortran, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command line interface, there are several graphical front-ends available.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

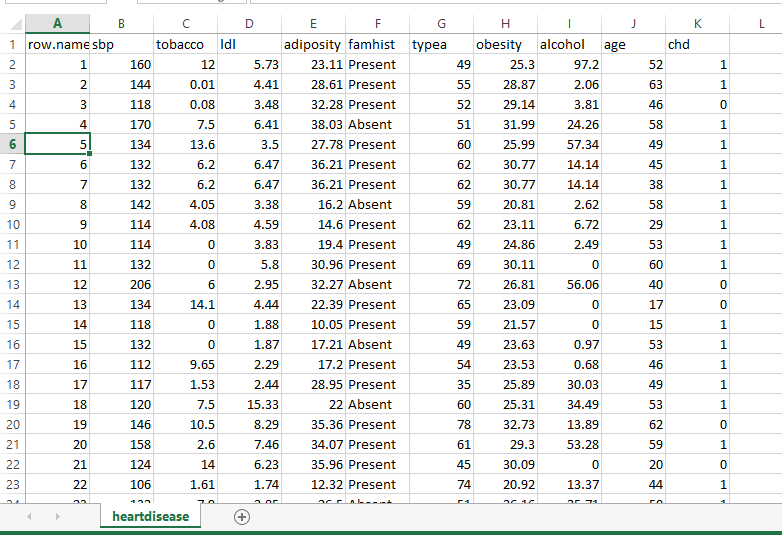
* An effective data handling and storage facility,
* A suite of operators for calculations on arrays, in particular matrices,
* A large, coherent, integrated collection of intermediate tools for data analysis,
* Graphical facilities for data analysis and display either on-screen or on hardcopy, and
* A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

**Description and Analysis of the Job**

These guidelines should be kept in mind to better understand the job at hand and should be followed to gain interesting knowledge from the dataset.

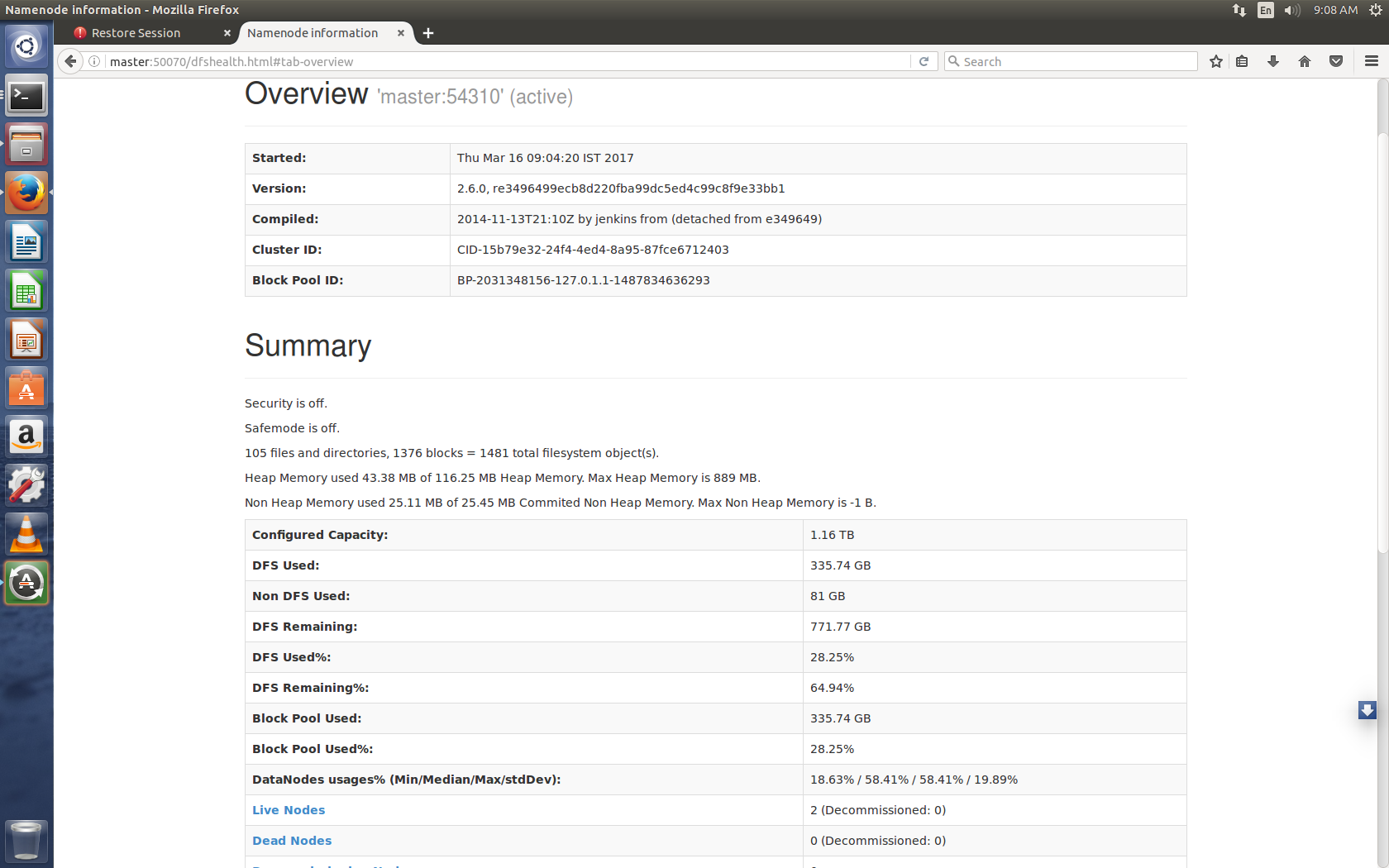
Selecting a dataset: The selection of dataset is based on the factors/fields that you would like to work on. As more and more data is being collected by various organizations/ companies, there are a number of datasets that you can work on.

Understanding the dataset: The dataset that you would like to work on should be studied thoroughly so as to understand the relationship between various variables or entities. Incomplete understanding of data may lead to false correlations and thus, wrong outcomes.



Storing the dataset on HDFS:

The dataset that is selected is stored on HDFS as HDFS can handle bigger data and keeps it in a distributed manner across various nodes.



Using R, analyze the dataset to gain useful outcomes:

Scoring of data is done using R and where modelization of data is done followed by linear regression.

**PROJECT:**

Description:

A sample dataset of males in a heart-disease high risk region of the western cape, South Africa. Following is the data provided within this dataset.

Format:

A data frame with 200 observations on the following 10 variables.

(1)sbp - systolic blood pressure

(2)tobacco - cumulative tobacco (kg)

(3)ldl - low density lipoprotein cholesterol

(4)adiposity - a numeric vector

(5)famhist - family history of heart disease, a factor with levels Absent Present

(6)typea - type-A behavior

(7)obesity - a numeric vector

(8)alcohol - current alcohol consumption

(9)age - age at onset

(10)chd - response, coronary heart disease

**Details:**

A sample dataset of a study conducted in males in a heart-disease high risk region of the Western Cape, South Africa. There are two controls per case of CHD. Many of the CHD positive men have undergone blood pressure reduction treatment and other programs to reduce their risk factors after their CHD event. In some cases the measurements were made after these treatments. This data are taken from a larger dataset, described in Rousseauw et al, 1983, South African Medical Journal.

**Major Objectives to achieve:**

* Exploration and Evaluation of Hadoop Distributed File System(HDFS)
* Study and Implementation of R
* Determining the probability of having a heart disease given the descriptors of the 200 patients by finding the risk of having heart disease based on different age groups.

**Requirements:**

**Hardware:**

Intel® CORE i3,i5,i7

RAM- 8 GB or more

Hard Disk- 1 TB

**Software:**

 Operating System- Ubuntu

Java 6 or above

Hadoop-2.6.0 or above

R-Studio (version 3.3.2)

**SCORING WITH R:**

This document is a guide to achieving a score with R through a practical implementation on South African Heart Disease dataset. It is divided into the following parts:

1 - Description of the study and data  2 - Preparation of the data and initial analyzes  
3 - Construction and validation of the score  
4 - Interpretation of the results

Scoring is a technique of hierarchization of the data which makes it possible to evaluate by a score or a score the probability that an individual responds to a solicitation or belongs to the target sought.

The score is usually obtained from the quantitative and qualitative data available on the individual (socio-demo data, purchasing behavior, previous answers, etc) to which a scoring model is applied.

In general, the modeling technique used is logistic regression. It helps to explain the belonging to a category from descriptors collected on a sample of population.

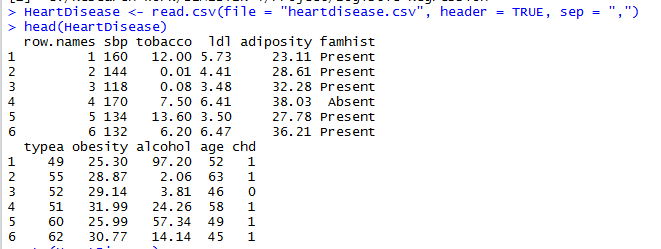
**Case study:**

The case study is based on a database containing 200 patients for whom it is desired to predict infarct exposure.

**Step 1:**

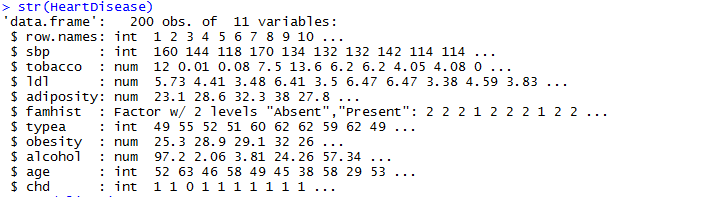
HeartDisease<- read.csv(file = "heartdisease.csv", header = TRUE, sep = ",")

head(HeartDisease)- Return the first or last parts of a vector, matrix, table, database or function.



**Step 2:**

str(HeartDisease)- compactly display the internal structure of an object in R.



**Description of the dataset:**

* Sbp: systolic blood pressure (systolic blood pressure)
* Tobacco: the amount of cumulative tobacco consumed (cumulative tobacco (kg))
* Ldl: blood cholesterol (low density lipoprotein cholesterol)
* Adiposity: adiposity
* Famhist: family history: Present if there was a family history of heart disease (Present, Absent)
* Typea: behavior type A (type-A behavior)
* Obesity: obesity
* Alcohol: current consumption of alcohol (current alcohol consumption)
* Age: age at time of heart attack (age at onset)
* Chd: variable coded response 1 if heart disease is present, 0 otherwise (response, coronary heart disease)

For modeling, we are interested in the last variable- CHD

**Definition of the purpose of modeling:**

We want to determine the probability of having a heart disease given the descriptors of the 200 patients **chd = f (obesity, age, family history, etc.).**

This example illustrates the use that can be made of Logistic regression, the formulation of the problem is generally the same, it can be for an insurance company, to determine the risk factors in order to provision (pricing), to determine the profiles of customers who are eager to a new commercial offer, and Macroeconomics, this approach is used to quantify country risk.

**The modeling approach:**

Like any good modeling approach, the construction of a good score is done by a succession of stages more or less fundamental according to the practitioners. Nevertheless, they all agree more or less to respect the following:

* Exploratory Analysis: What's in the Dataset? Are there many missing values?
* Check the link between the descriptors and the variable to be explained
* Identify important and redundant predictors to make a parsimonious model (very important when predicting)
* Estimate the model on a learning sample
* Validate the model on a test sample and build the quality indicators of the model
* Compare several models and retain the most appropriate model according to the objective of the study

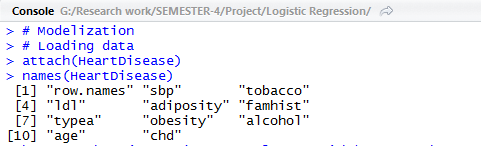
We will follow these steps and solve the problem posed by our case study.

**Step 3: Modelization**

**Loading data**

attach(HeartDisease)- the database will be attached to R search path so as to access objects in the database by their names.

names(HeartDisease)-function to get or set the names of an object.



**Step 4:**

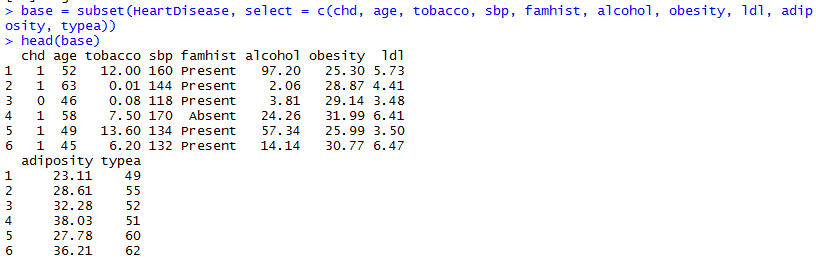
Base contains basic R functions like arithmetic etc.

Subset returns subsets of vectors, matrices or data frames which meet conditions.

Select is used to include all those attributes that satisfy he given profile.

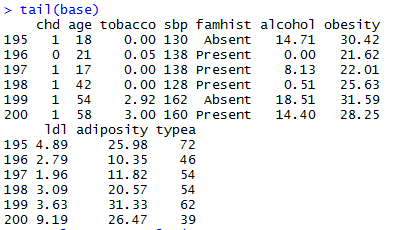
base=subset(HeartDisease, select=c(chd, age, tobacco, sbp, famhist, alcohol, obesity, ldl, adiposity, typea))

head(base)



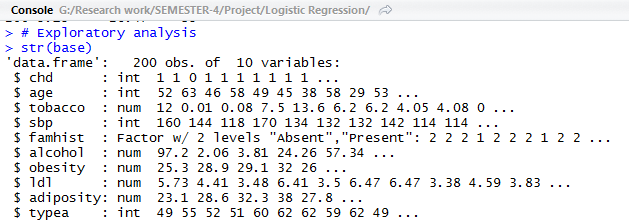
**Step 5:**

tail(base)



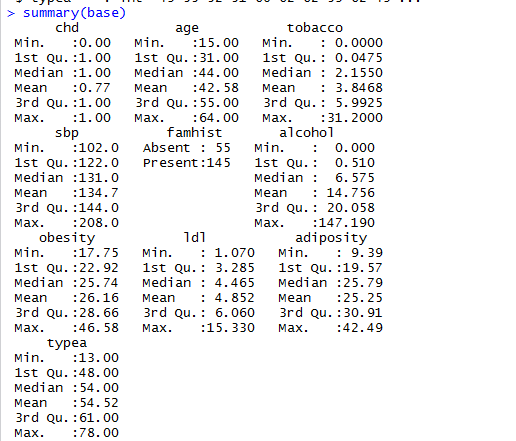
**Step 6: Exploratory analysis**

str(base)



**Step 7:**

summary(base)- Generic function used to summarize the results of various models fitting functions.

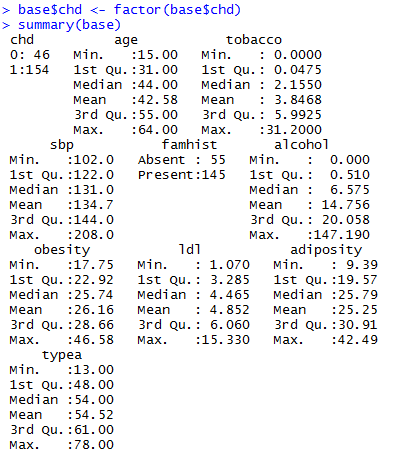


**Step 8:**

It is noted that the target variant *chd* has been treated as a numerical variable: This is re-mediated by transforming it into factor type. Factor is stored as a vector of integer values with a corresponding set of char values to use when the factor is displayed. Factor function is used to create a factor.

base$chd<- factor(base$chd)

summary(base)



**Step 9: Searching for more relevant explanatory variables**

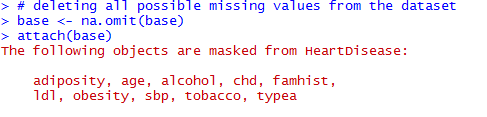
We will try to identify the possible correlations between the descriptors

But before that: We will delete all possible missing values ​​from the dataset

base<- na.omit(base)

Let us analyze the distributions

attach(base)



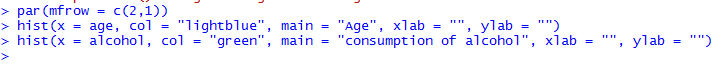
**Step 10:**

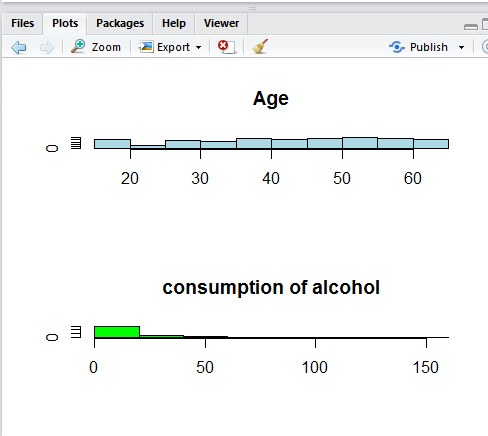
Par is used to combine plots and mfrow is used to create matrix of n rows n columns that are filled in by row. mfcol-acc to cols.

par(mfrow = c(2, 1))

hist(x = age, col = "lightblue", main = "Age", xlab = "", ylab = "")

hist(x = alcohol, col = "red", main = "Consommationcour. d'alcool", xlab = "", ylab = "")





**Step 11:**

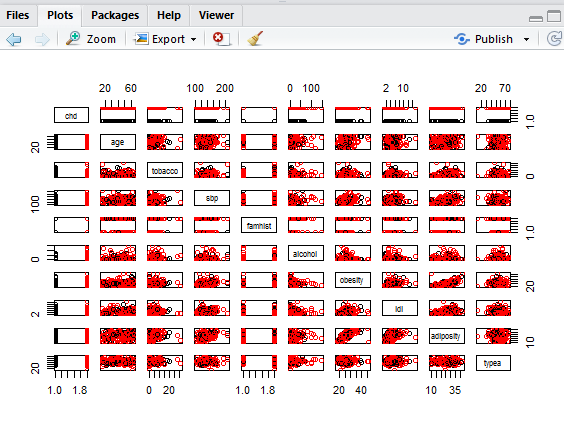
Another analytical tool is also to create point clouds for all variables. It is possible to color the points according to the target variable

par(mfrow = c(1, 1))

pairs(base, col = base$chd)



Pairs is used to produce a matrix of scatter plots.



**Step 12:**

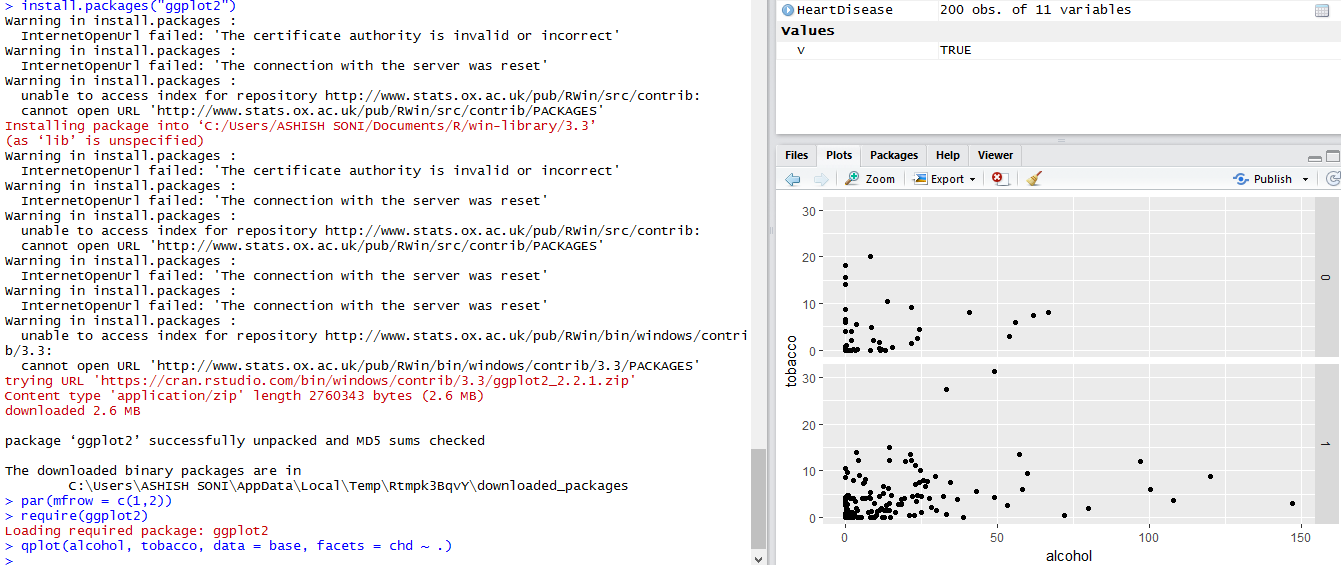
More graphics are possible through the *ggplot* package   
for example:

par(mfrow = c(1, 2))

require(ggplot2)

## Loading required package: ggplot2

qplot(alcohol, tobacco, data = base, facets = chd ~ .)- this function can be used to create various plot types.



**Step 13:**

#### Discretize or not? \*

This is a frequent question in exploratory analysis. The discretization of continuous variables.

In the dataset, the variables:

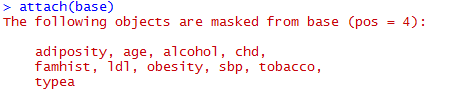
* age
* Tobacco
* Sbp
* Alcohol
* Obesity
* Ldl
* Adiposity
* Typea   
  are potentially discretized.

The age variable is usually the easiest to cut. The cardiac problems do not affect in the same way according to the ages  
We choose to discretize the age and tobacco variables by distinguishing between small, medium and heavy smokers.

detach(HeartDisease)- detaches the database i.e. remove it from the search path of available R objects.

attach(base)



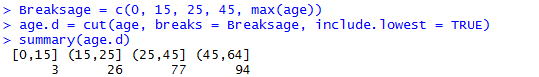


**Step 14:**

Breaksage = c(0, 15, 25, 45, max(age))

age.d = cut(age, breaks = Breaksage, include.lowest = TRUE)- cut divides a continuous variable into a categorical variable.

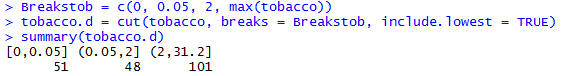
summary(age.d)



Breakstob = c(0, 0.05, 2, max(tobacco))

tobacco.d = cut(tobacco, breaks = Breakstob, include.lowest = TRUE)

summary(tobacco.d)



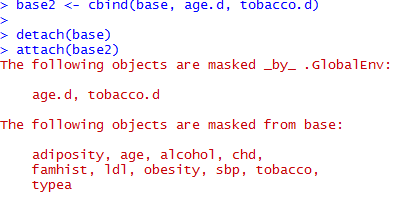
**Step 15:**

A division is relevant only when the terms of the target variable are correctly represented

base2 <- cbind(base, age.d, tobacco.d)- cbind function takes a vector, matrix or data-frame and combine by cols or rows.

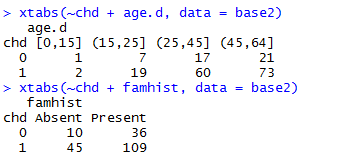
detach(base)

attach(base2)

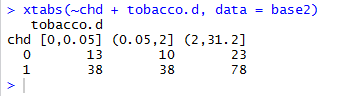


xtabs(~chd + age.d, data = base2)- creates a contingency table from cross-classifying factors using a formula interface.

xtabs(~chd+famhist, data=base2)



xtabs(~chd+tobacco.d, data=base2)



The category of less than 15 years is not representative at all in the sample. The under 25s either.

Half of people over the age of 45 have a heart condition.   
We can think of a form of inheritance in heart problems in view of cross-breeding.   
It is clear that smoking has a real influence on heart problems, because there is a significant proportion of people affected whatever their amount of tobacco consumed.

These initial analyzes indicate that:

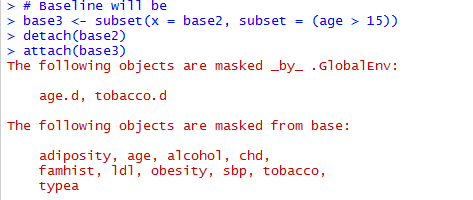
* It is not very useful to keep individuals under the age of 15 in the sample because the model being developed would not be calibrated to predict their likelihood of developing heart disease if age plays a role Therefore.
* Some of the results described here descriptively need to be confirmed in a modeling phase.

**Step: 16**

base3 <- subset(x = base2, subset = (age > 15))

detach(base2)

attach(base3)



**Step 17:**

base3 <- subset(base3, select = -c(age, tobacco))

set.seed(111)-to set the seed of random number generator useful for creating simulations or random objects that can be reproduced.

d = sort(sample(nrow(base3), nrow(base3) \* 0.65))- nrow is to return the no of rows and cols present.

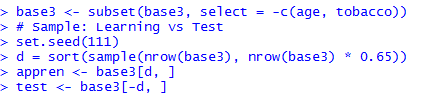
d- dataframe and sample allows to determine the size required to detect and affect.

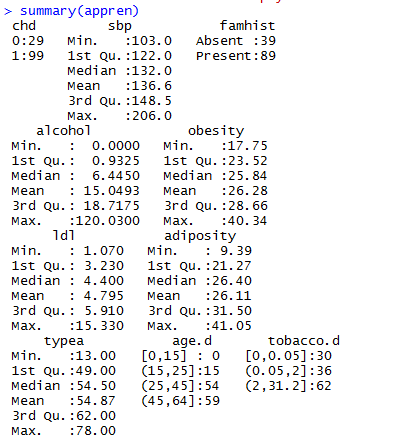
Sample-

appren<- base3[d, ] -

test<- base3[-d, ]

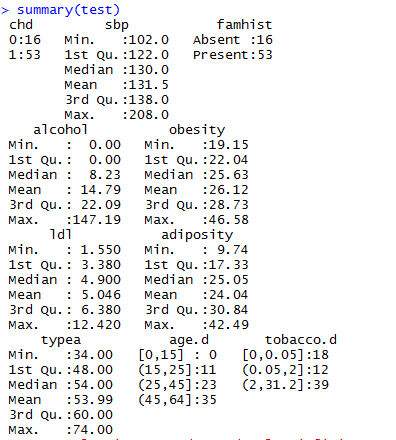
summary(appren)





**Step 18:**

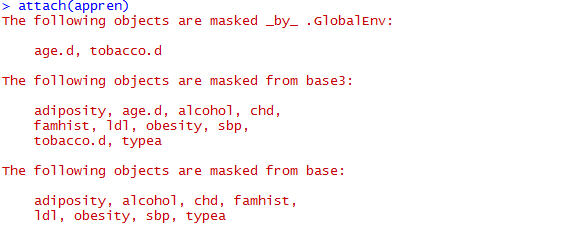
summary(test)



**Step 19:**

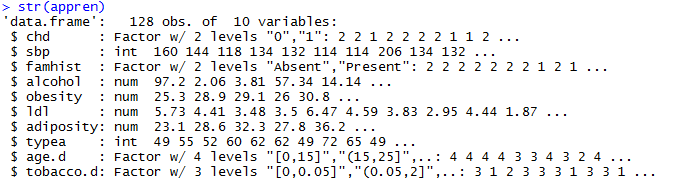
Before embarking on the construction of models, let us briefly describe our learning sample

attach(appren)



**Step 20:**

str(appren)



**Step 21:**

**Construction of models**

*We want to model chd = f (explanatory variables)*

In real life, many more variables are available, and a criterion that selects the variables to be introduced into the model must be used. We do it here for purely illustrative purposes, we could do without in this example. We also leave aside the analyzes of the interactions between variables which constitute a full stage of the analysis because the reading of the results is not done in the same way.

*What variables to introduce?*

But if we were to retain only one, it would be the following: \*\* A model with few variables will be more easily generalizable in terms of robustness (Occam's razor principle) \*\*

Several ways to select:

* Backward: We start from all the available variables and we remove as the non-significant variables
* Forward: The opposite of Backward
* Both: in both directions (stepwise equivalent under SAS) In general, the Akaike criterion (AIC) or Scharwz (BIC) criterion is the one we want to minimize: These almost equivalent criteria translate The complexity of the model as we add variables (therefore parameters to be estimated) In our example, let's implement the different selections

str\_constant<- "~ 1"

str\_all<- "~sbp+famhist+alcohol+obesity+ldl+adiposity+typea+age.d+tobacco.d"

* Forward

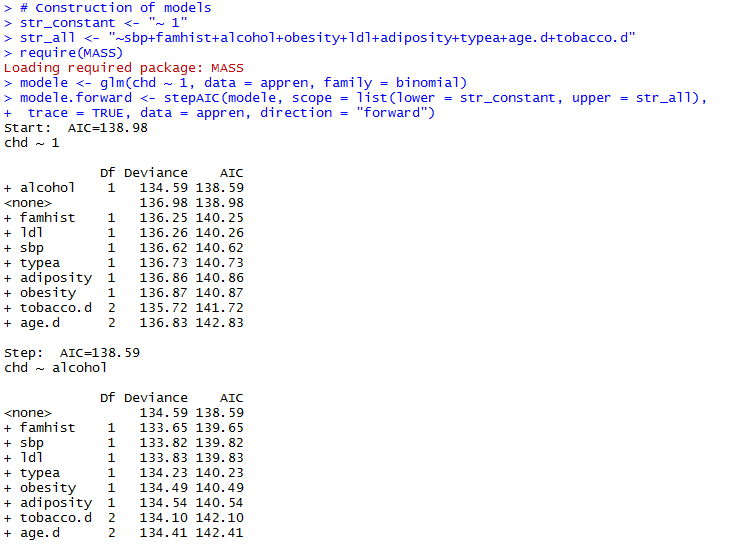
require(MASS)

## Loading required package: MASS

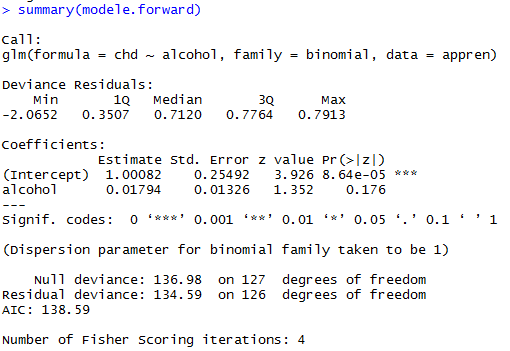
modele<- glm(chd ~ 1, data = appren, family = binomial) –glm is gor generalized linear models, by giving symbolic description of linear predictor and error distribution.

modele.forward<- stepAIC(modele, scope = list(lower = str\_constant, upper = str\_all),

trace = TRUE, data = appren, direction = "forward")



summary(modele.forward)



**Step 22:**

Let us analyze the algorithm made by R. In the basic model, AIC = 367 It adds tobacco consumption as an explanatory variable and calculates the AIC in ascending order of the variables introduced: AIC = 367; *It is the best model at this stage*

It tests all the models with two variables without questioning the consumption of tobacco since already selected: it retains the family antecedent   
etc...

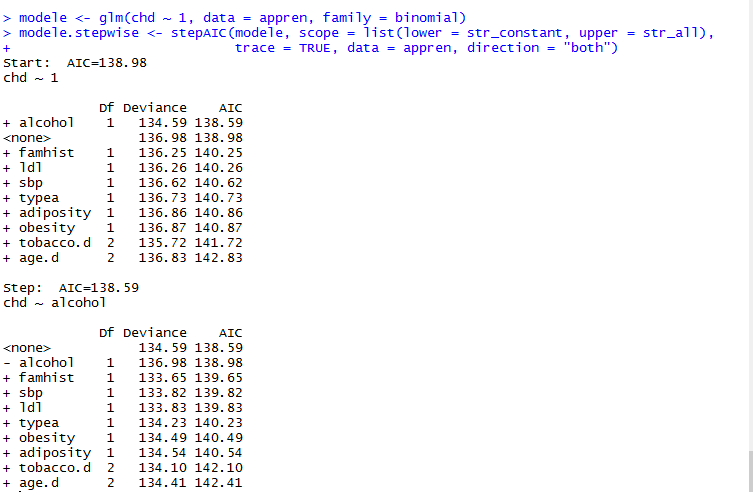
The model that proposes in the end Call: **Glm (formula = chd ~ tobacco.d + famhist + age.d + typea + ldl + obesity + sbp, family = binomial, data =** appren **)** retains all variables with the exception of alcohol consumption The consumption of tobacco as had been detected in the "

* BOTH: Stepwise: This is the most effective and most used method in practice. Nevertheless, there are a variety of other methods, such as selections based on a combination of Wald's Forward + Statistics on the coefs. But this is a little beyond the scope of what is being sought here.

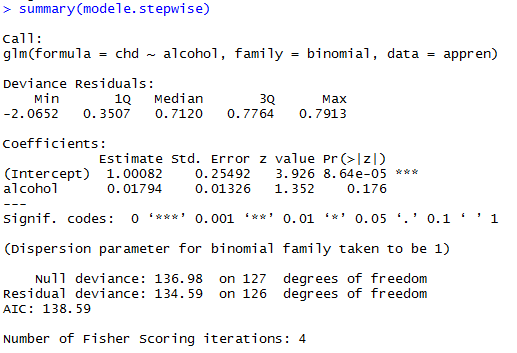
modele<- glm(chd ~ 1, data = appren, family = binomial)

modele.stepwise<- stepAIC(modele, scope = list(lower = str\_constant, upper = str\_all),

trace = TRUE, data = appren, direction = "both")- stepAIC performs stepwise model selection by AIC



summary(modele.stepwise)



**Step 23:**

Explaining the approach as we did *Backward* would be a bit complicated, so let's look at the final selection made by algorithm:

**Glm (formula = chd ~ tobacco.d + famhist + age.d + typea + ldl + obesity + sbp, family = binomial, data = appren)** Let the same variables as in *forward* selection .

**Modelization**

Before continuing, I propose to build ourselves our function to carry out a logistic regression under R. This only in the bit to lighten the writing of our models. The function used for logistic regression is simply the glm function of R.

logit = function(formula, lien = "logit", data = NULL) {

glm(formula, family = binomial(link = lien), data)

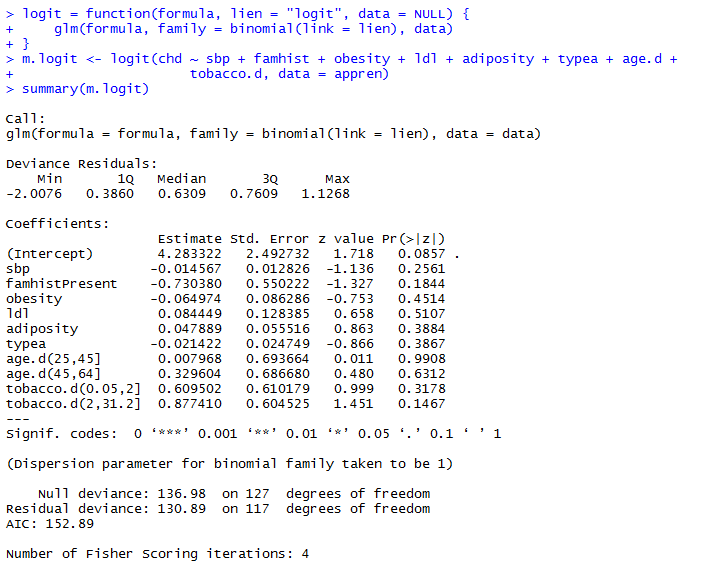
}

We introduce all the variables in our model except the alcohol conso:

m.logit<- logit(chd ~ sbp + famhist + obesity + ldl + adiposity + typea + age.d +

tobacco.d, data = appren)

summary(m.logit)



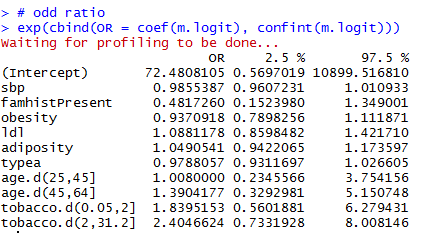
**Step 24:**

* Interpretation of coefficients

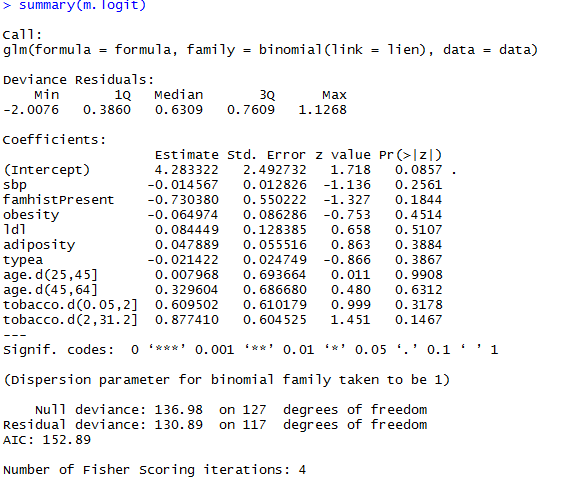
By default, R does not give odds-ratios. But it is easily calculated by remembering that an odd-ratio is nothing more than the estimate of the coefficients of regression. The Odd-ratio and their confidence intervals can be obtained by

exp(cbind(OR = coef(m.logit), confint(m.logit)))

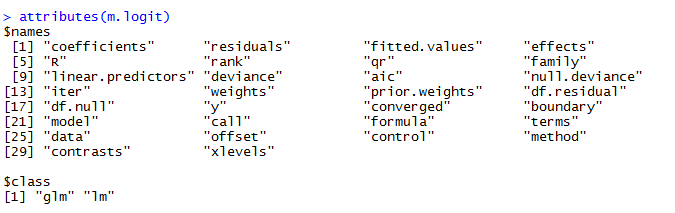
## Waiting for profiling to be done...



summary(m.logit)



attributes(m.logit)



**Step 25:**

**Validation of the model: Indicators of quality and robustness**

After obtaining a model, it is necessary to diagnose the regression in order to validate or not the model.

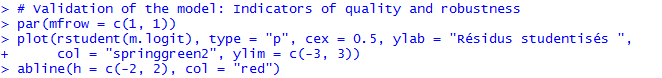
From this point of view, residue analysis is very important. It is important to note that in logistic regression, one is mostly interested in residues of deviance. They generally take values ​​that oscillate between -2 and 2. A plot index is usually constructed to detect outliers (outside rows)

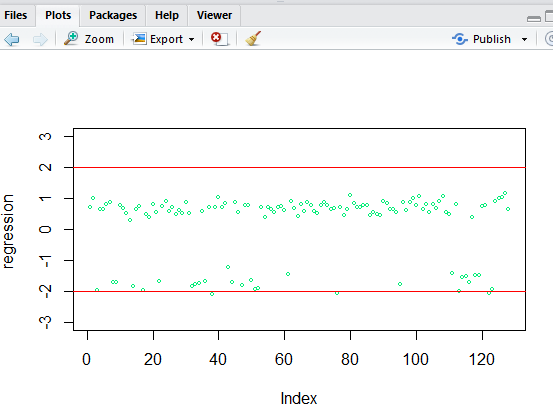
par(mfrow = c(1, 1))

plot(rstudent(m.logit), type = "p", cex = 0.5, ylab = "Résidusstudentisés ",

col = "springgreen2", ylim = c(-3, 3))

abline(h = c(-2, 2), col = "red")





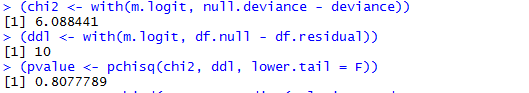
**Step 26:**

There is also general interest in the deviance of the model: the likelihood ratio tests and the calculation of the p\_value with the degree of freedom difference between the reduced model at the cste and the chosen model gives the overall significance of the model .

(chi2 <- with(m.logit, null.deviance - deviance))

(ddl<- with(m.logit, df.null - df.residual))

(pvalue<- pchisq(chi2, ddl, lower.tail = F))



**Step 27:**

We will try to validate now on the test sample that we have previously defined;

Here are the steps we will follow to validate our model

On the learning sample and on the test sample:

* A confusion matrix is ​​calculated: and therefore an error rate
* The air is evaluated under the ROC curve

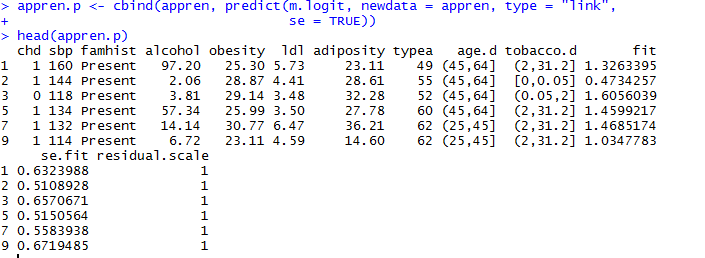
But before, to predict with R, a function \* predict () \* whatever the model developed and the package used 

So let:

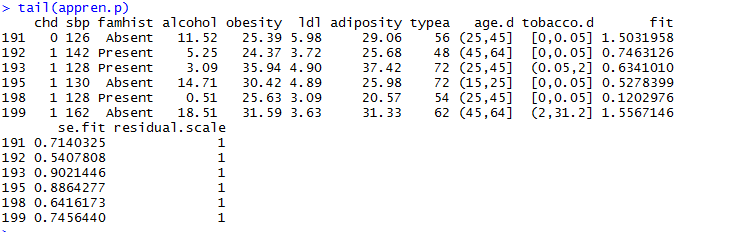
appren.p<- cbind(appren, predict(m.logit, newdata = appren, type = "link",

se = TRUE))

head(appren.p)



tail(appren.p)



**Step 28:**

**ROC curve *construction* : *receiver operating characteristic***

This curve, or rather the area under it, represents the sensitivity / specificity of the model. A model is good if the positive (1) were   
predicted positive and the 0 were predicted 0.

Generally, we are interested in both the shape of the curve and the area under it: 1-> Model Ideal, 0.5 -> Random pattern;

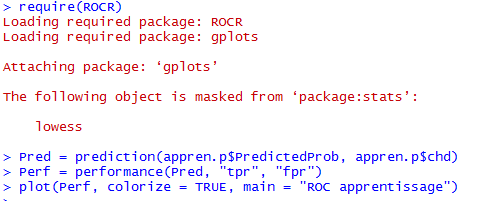
Principle of the ROC curve: if the test gives a numerical result with a threshold t such that the prediction is positive if x> t, and the prediction is negative if x <t, then as t increases:

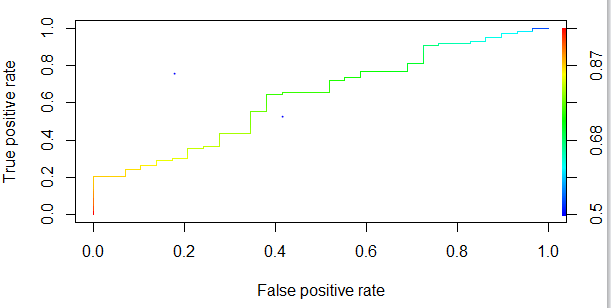
* The specificity increases.
* But the sensitivity diminishes.

The ROC curve represents the change in sensitivity (true positive rate) as a function of 1 - specificity (false positive rate) when the threshold t is varied.

It is an increasing curve between the point (0,0) and the point (1,1) and in principle above the first bisector. A random prediction would give the first bisector. The better the prediction, the more the curve is above the first bisector. An ideal prediction is the horizontal y = 1 on] 0,1] and the point (0,0). The area under the ROC (Area Under the Curve) curve gives an indicator of the quality of the prediction (1 for an ideal prediction, 0.5 for a random prediction).

require(ROCR)





**Step 29:**

To have the area under the curve,

perf<- performance(Pred, "auc")

perf@y.values[[1]]

We will put the two ROC curves side by side (learning and tests)

Predtest = prediction(test.p$fit, test.p$chd)

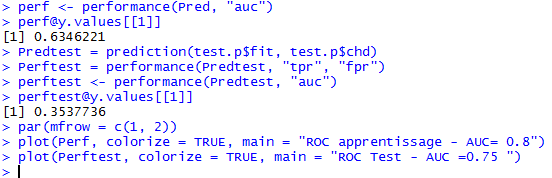
Perftest = performance(Predtest, "tpr", "fpr")

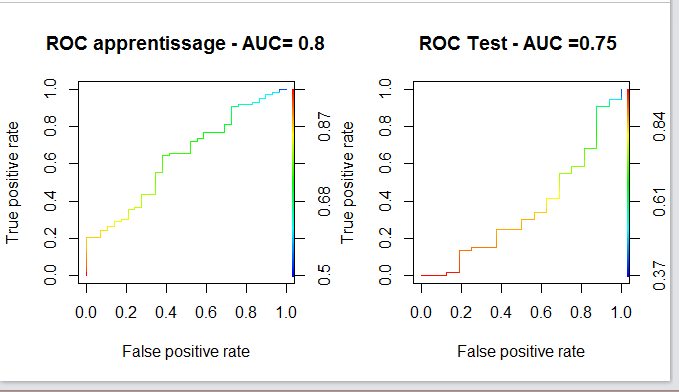
perftest<- performance(Predtest, "auc")

perftest@y.values[[1]]

par(mfrow = c(1, 2))

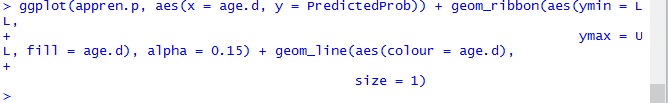
plot(Perf, colorize = TRUE, main = "ROC apprentissage - AUC= 0.8")

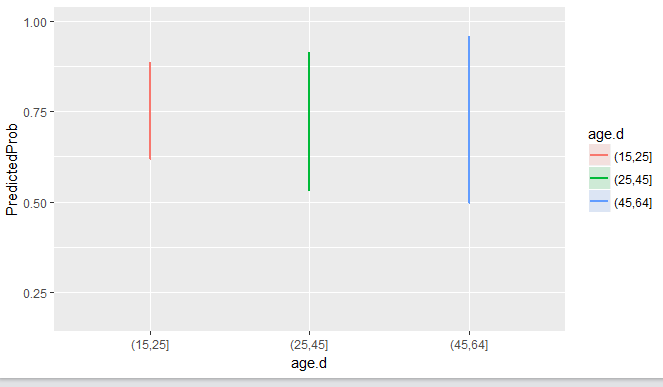




**Step 30:**

ggplot(appren.p, aes(x = age.d, y = PredictedProb)) + geom\_ribbon(aes(ymin = LL, ymax = UL, fill = age.d), alpha = 0.15) + geom\_line(aes(colour = age.d), size = 1)





**INDUSTRY APPLICATION**:

In today’s world, there are huge amounts of data available in each industry. The healthcare industry deals with a lot of data daily which can be utilized to better monitor or predict the risk of heart disease in the general public. So, analyzing those datasets can help in preventing and taking better measures to avoid heart diseases which can be fatal to a person’s health. Therefore, it helps to predict which age groups are prone to heart diseases due to various factors and provide the necessary diagnosis which can help in providing better solutions.

**CONCLUSION:**

The use of Data Mining Techniques helped us determine that, analyzing patient information, based on data classification and prediction analysis, we identified that some variables can be used as proper predictors of heart disease.

* The identified variables: sbp, ldl, chd, and famhist etc are the predictors for heart disease.
* Various histograms were obtained from the given dataset that indicated the relationship of various attributes.
* Modelization and discretization was done on the basis of the available dataset and it was found that people under the age of 15 were not relevant for the analysis.
* Test base was made too so as to check which model was better – forward AIC, backward AIC or both AIC.
* ROC curve was made to validate the model.
* It was found that people between the ages of 45-64, based on the dataset, are more prone to a heart disease than people between the ages of 25-45 and 15-25.

**REFERENCES:**

* Source of the dataset:

Rousseauw, J., du Plessis, J., Benade, A., Jordaan, P., Kotze, J. and Ferreira, J. (1983). Coronary risk factor screening in three rural communities, South African Medical Journal 64: 430–436.

* <http://hadoop.apache.org/>
* Padhy, Rabi Prasad. "Big data processing with Hadoop-MapReduce in cloud systems." International Journal of Cloud Computing and Services Science 2.1 (2013): 16.
* Fan, J., Han, F. and Liu, H., 2014. Challenges of big data analysis. National science review, 1(2), pp.293-314.