Implementation of Data Warehouse for Rankings of Universities in USA

Sunil Kumar Vuppalapati| X17153590

MSC. DATA ANALYTICS | NATIONAL COLLEGE OF IRELAND

SUBMITED TO: DR.ANU SHANI

INDEX

Introduction:

In the era of data originating from both online and offline resources leading to innovative concepts like bigdata, data accumulation, storage and retrieval plays a prominent role in building a data warehouse as required by the organisation. Data Warehouse comprises of multiple data types like historical data, current market scenarios, target population decision analysis etc. Data warehouse features like time-variant, non-volatility aids in developing a reliable analytical reporting database for successful organisational business requirements. In this project my data warehouse is a combination of collective data sets from different sources developed into a data warehouse of education institutions with different parameters listed and ranked respectively serves as a reliable resource in selecting the correct universities based on the student interest.

Collection of datasets and their sources:

The Selection of the datasets are important requirement as they are used to build the data warehouse, so that they provide enough and accurate information which meet our requirements. In this project I took 7 datasets from 6 different sources, of which six are structured and one unstructured data.

Data Source 1:

My first data set is CWUR (Centre for World University Rankings) where I collected the data from the table using an R code, which comprises national rankings, ranking for citations, publications and overall score for each university.

Source URL: http://cwur.org/2017.php

Data Source 2:

My second data source is from National Science Foundation where I collected two data sets which has data related to number of graduates per year and R&D expenditure per year for each university.

Source URL : https://ncsesdata.nsf.gov/profiles/site?method=rankingBySource&ds=herd

Data Source 3:

My third data source is from Times Higher Education where I collected the table data using data miner tool. This data includes fees, room and board fees, pay after graduation, and scores of variables like resources, student staff engagement, and opportunities.

Source URL: https://www.timeshighereducation.com/rankings/united-states/2018#!/page/0/length/25/sort_by/rank/sort_order/asc/cols/stats

Data Source 4: -

My fourth data source is from Webometrics website, from which I acquired table data using R code and this data provides information of different rankings of universities like Impact rank, Openness rank, Excellence rank.

Source URL: http://www.webometrics.info/en/North_america/United%20States%20of%20America

Data Source 5:

Forbes website is the source from which my 5th dataset is originated from. This data set comprises of attributes like Financial Aid and type of university.

Source Url: https://www.forbes.com/top-colleges/list/

Data Source 6:

My last data source is from Twitter where I have done sentiment analysis for each university. Using Twitter API, I have executed R code for the output on positive and negative scores.

Tools which are used in project:

1)SQL Server Integration Services (SSIS)

2)SQL Server Management Services (SSMS)

3)SQL Server Analysis Services (SSAS)

4)R-Studio

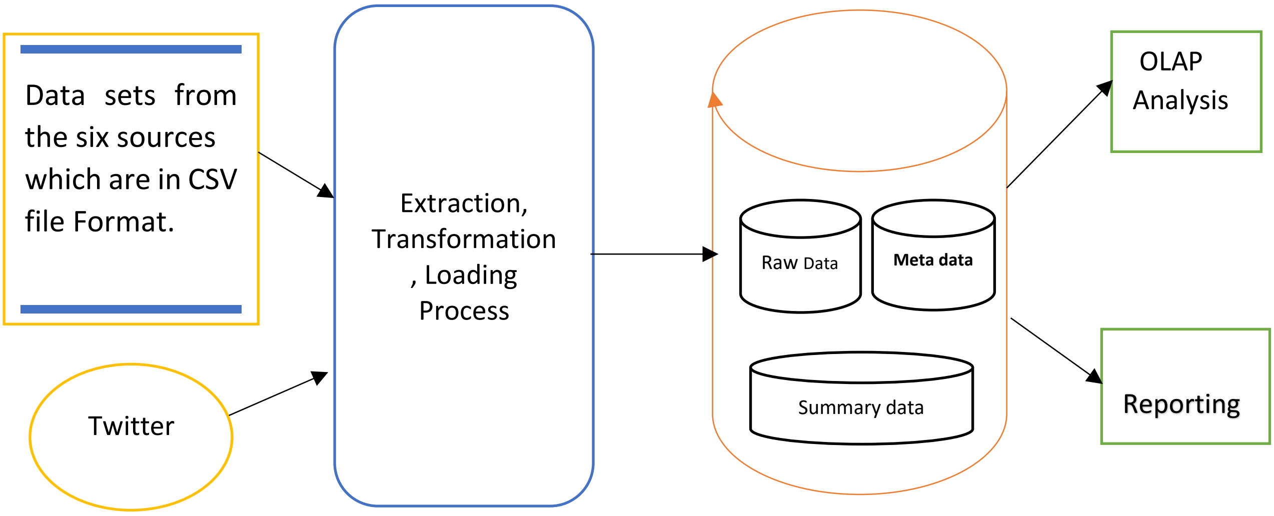
5)Microsoft-Excel

6)Tableau

Implementation of Data Warehouse Architecture:

Selecting implementation approach for the data warehouse in the available approaches like Ralph Kimball, Bill Inmon and mixed approach is crucial, as it must match with our requirements. In this project, I used Kimball approach over Inmon approach because it is easy and appropriate for my data as I am creating data marts from the different sources and followed by ETL in staging. Later transferring into universities data warehouse and then finally we will present the analysis and reporting data to the students. Another advantage of Kimball approach made the building of data warehouse in a fast time and it consists of many dimensions and facts which allows the students to analyse and find the desired universities.

The architecture which I followed for creating the data warehouse is represented in the below picture.


Data sets from the six sources which are in CSV file Format.
Twitter
Extraction, Transformation , Loading Process
Raw Data
Meta data
Summary data
OLAP Analysis
Reporting

Data warehouse architecture

Data Model of University Data warehouse:

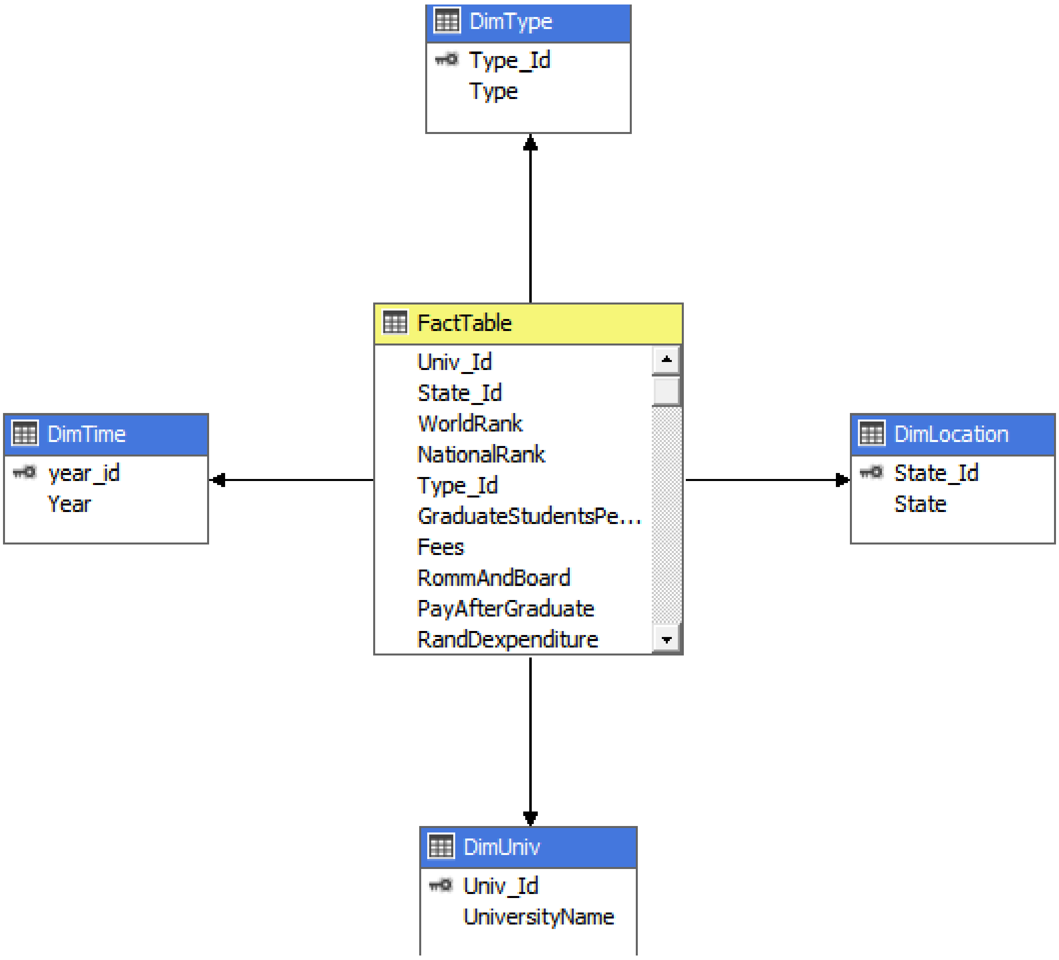
In the implementation of the University Rankings in USA data warehouse I used STAR schema over snow flake. The star schema is easy to make dimension tables and connect to the fact table and we are not using any normalised dimensions, so the star schema is best model with the following advantages over other schemas

- Simple structure which makes easy implementation of data warehouse
- Easy queries because of it directly connected to fact table
- Widely supported by the business intelligence tools

A start schema generally consists of a fact table in the centre and surrounded by the dimensional tables

Design of My Project:

The design of the My Project which consists of one fact table and four-dimension tables which are connected through the foreign key references in the fact table. All dimension tables are defined for business analysis based on the values which they stored.



STAR schema of USA Universities data warehouse

Fact Table:

The fact table which consists of all key values of dimensions and measures such as number of graduates, national rank, fees, room and board fee, financial aid, pay after graduation, citations, resources, impact rank, openness rank, excellence rank and other fields like sentiment positive, negative score and overall score. With these facts and measures we find the case studies like what is the average return of investment, Best universities for research-oriented graduation and other two cases as explained in the case studies.

Dimensions:

From the above schema DimUniv, DimLocation, DimTime, DimType are the dimensions and the FactTable is fact table which are connected through the unique primary which are defined in the dimension tables.

DimUniv:

It contains the name of the university this attribute helps in separating data based on the university name and its ranks.

DimLocation:

It contains the name of the 'State' this attribute helps in separating the universities based on the locations.

DimTime:

It contains the date of the 'Year' this attribute helps in the finding the amount of R and D amount data and number of graduates for the year and ranking of universities for that year.

DimType:

It contains the Type column through which it represents the type of university and this attribute helps in finding what type of universities more helpful for students.

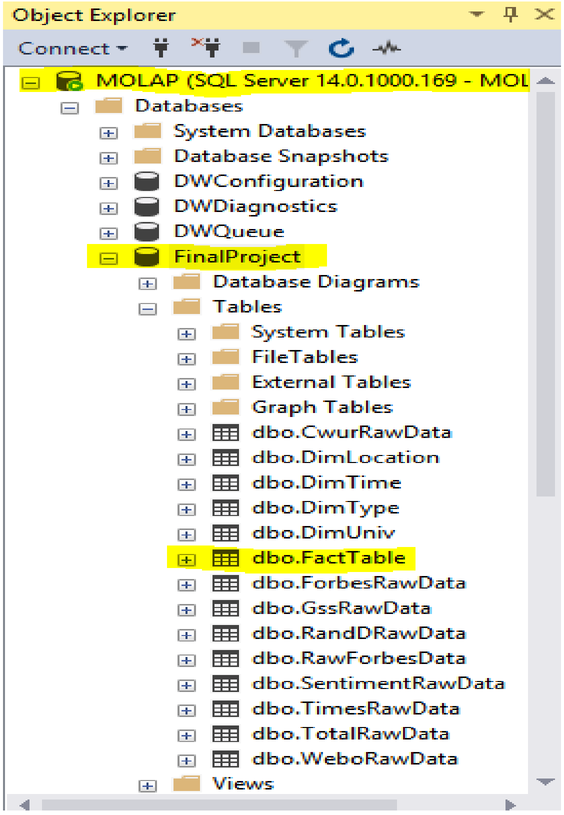
Extraction, Transforming, and Loading Process:

ETL (Extract, Transform and Load) Is the key process where we are acquiring data from different sources and transforming into desired format which is easily load to data warehouse.

The ETL process in this project is starts with the extraction of the data from the sources. For the sources CWUR and Webometrics I used R code to do web scrapping to get the table data. [R code is mentioned in Appendix A]. To fetch data from the sources THE (Times Higher Education) and Forbes I use data mining tool called as Data Miner. The other datasets R&D data and the Graduates student's data in CSV format are downloaded directly from the National Science Foundation website and the unstructured data source is taken from the Twitter API using httr, twitteR, plyr, stringr packages in R [R code is in Appendix B].

After extraction of required data from the sources we need to clean the data so that the information will be accurate, and it has to match with the database tables. for that we need to clean the data like replacing null values removing the unnecessary symbols and filtering the columns which we are going to use in our data warehouse and finally removing the rows or columns for faster loading into the data warehouse. For transforming and cleaning I used the R code to make sure that data from all sources should match for that I have taken CWUR as a main dataset and with that I compared all the data sets by the common column university name[R code is included in Appendix C] and then I used Microsoft Excel for further cleaning naming the columns.

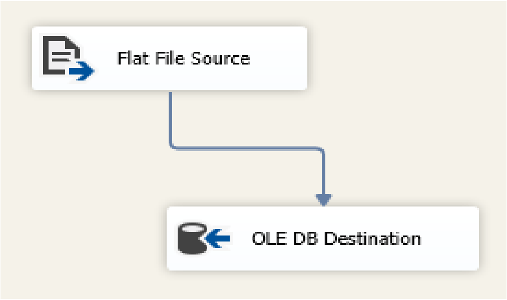
The final part of ETL is loading, for loading the data into to the data warehouse we use tools SSIS and SSMS. In SSMS I created a database named as Final Project where I am going to save all the data in the form of tables and by using SQL queries. I created the dimension tables and the fact table the below picture shows the database and the tables which I created to store the data initially.

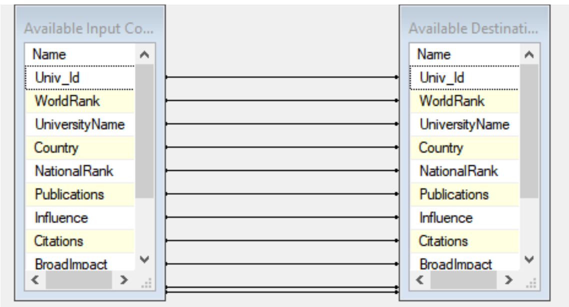


```
MOLAP (SQL Server 14.0.1000.169 - MOL
  Databases
    System Databases
    Database Snapshots
    DWConfiguration
    DWDiagnostics
    DWQueue
    FinalProject
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.CwurRawData
        dbo.DimLocation
        dbo.DimTime
        dbo.DimType
        dbo.DimUniv
        dbo.FactTable
        dbo.ForbesRawData
        dbo.GssRawData
        dbo.RandDRawData
        dbo.RawForbesData
        dbo.SentimentRawData
        dbo.TimesRawData
        dbo.TotalRawData
        dbo.WeboRawData
      Views
```

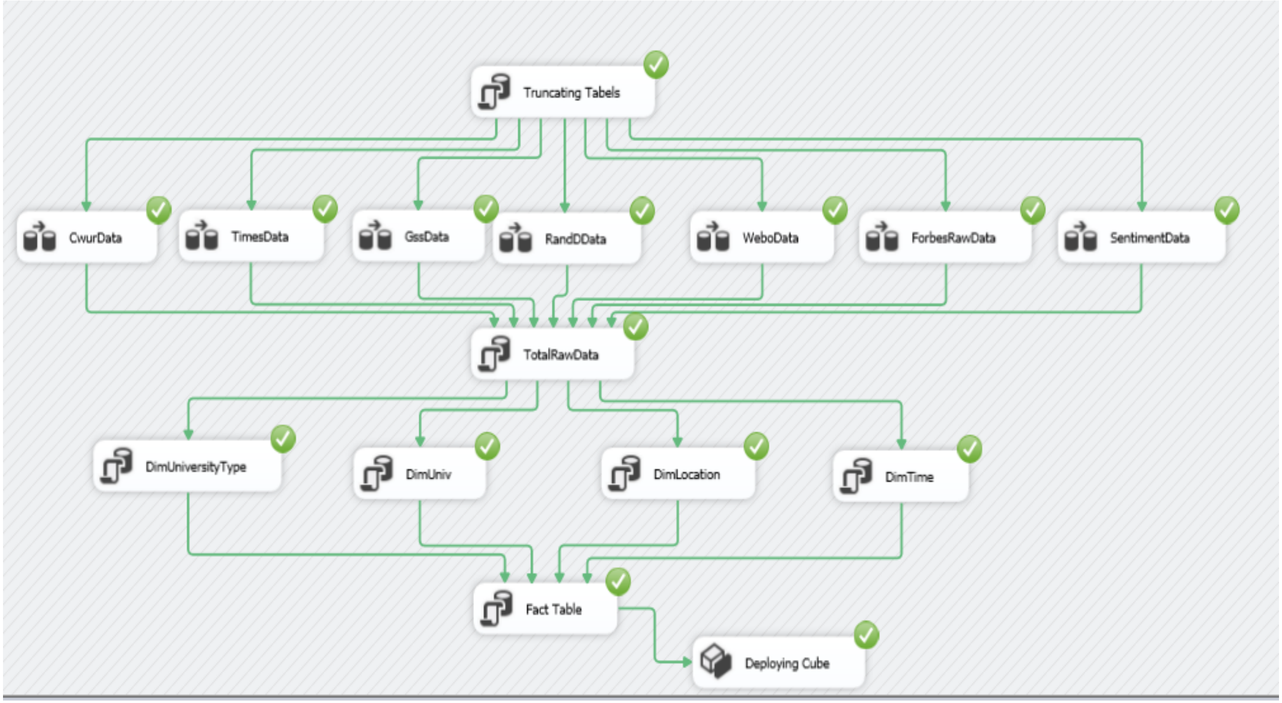
Database in SSMS for the project

To populate the data, I am using SSIS, as it can load data from multiple sources into a single flow which makes the work easy and reduces the operational time. In the data importing from CSV files to OL EDB destination we use a data flow task in SSIS. In that we can able to transfer data from csv to database by using necessary mappings with the table in the database to avoid the errors while populating data. We are using truncation of tables wherever it is necessary so that while updating the new data it won't give errors. In final we will segregate data into separate tables as dimensions and fact tables depend on the requirements.



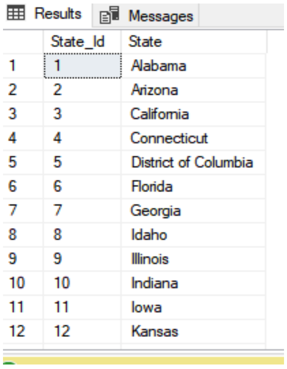


Flat File Source Mappings



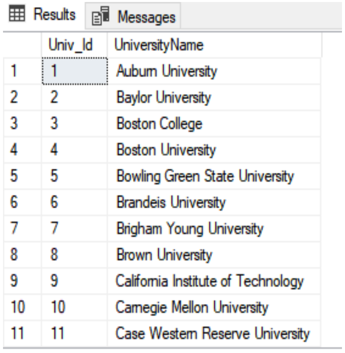
Work flow of SSIS

The above figure shows the control flow in SSIS. The flow starts with the truncation of the tables with that new data can update whenever it is needed. It is followed by the data flow tasks where we are loading data from CSV files to OLEDB destination and later we are populating characteristic data into the dimension tables using SQL queries using SQL execute task. In the last step the data which it consists of measures moved into fact table. The green ticks in the above indicates that the task is completed successfully.



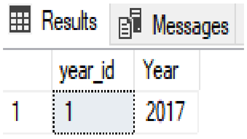
	State_Id	State
1	1	Alabama
2	2	Arizona
3	3	California
4	4	Connecticut
5	5	District of Columbia
6	6	Florida
7	7	Georgia
8	8	Idaho
9	9	Illinois
10	10	Indiana
11	11	Iowa
12	12	Kansas

DimState



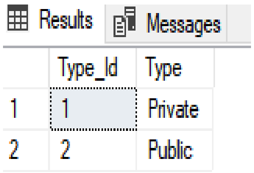
	Univ_Id	UniversityName
1	1	Auburn University
2	2	Baylor University
3	3	Boston College
4	4	Boston University
5	5	Bowling Green State University
6	6	Brandeis University
7	7	Brigham Young University
8	8	Brown University
9	9	California Institute of Technology
10	10	Carnegie Mellon University
11	11	Case Western Reserve University

DimUniv



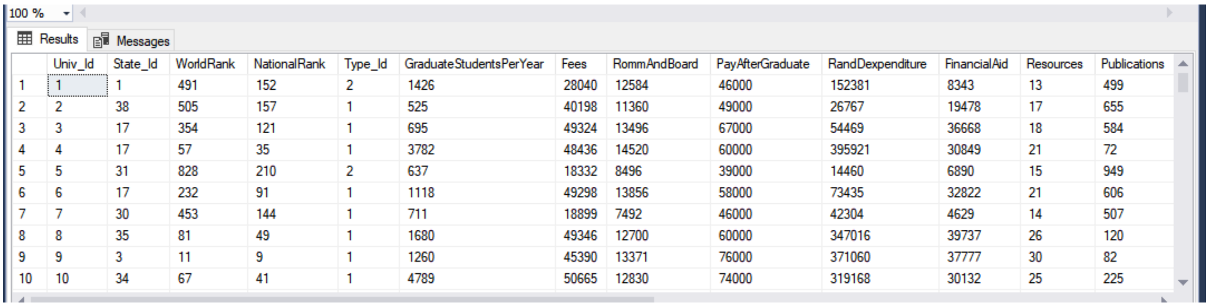
	year_id	Year
1	1	2017

DimTime



	Type_Id	Type
1	1	Private
2	2	Public

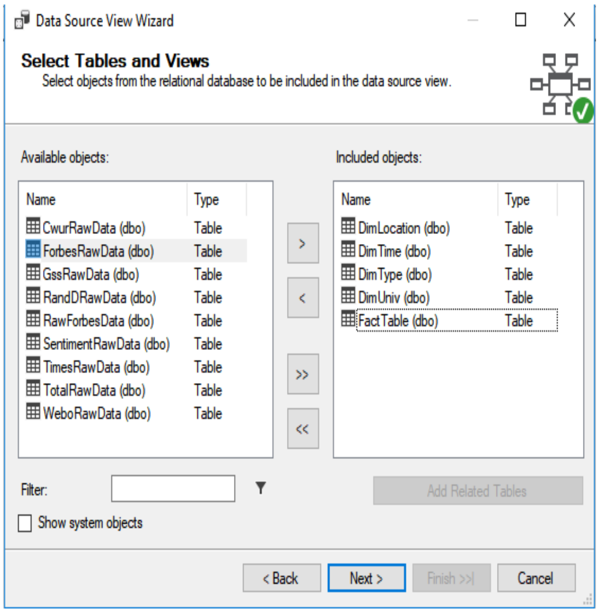
DimType



	Univ_Id	State_Id	WorldRank	NationalRank	Type_Id	GraduateStudentsPerYear	Fees	RommAndBoard	PayAfterGraduate	RandDexpenditure	FinancialAid	Resources	Publications
1	1	1	491	152	2	1426	28040	12584	46000	152381	8343	13	499
2	2	38	505	157	1	525	40198	11360	49000	26767	19478	17	655
3	3	17	354	121	1	695	49324	13496	67000	54469	36668	18	584
4	4	17	57	35	1	3782	48436	14520	60000	395921	30849	21	72
5	5	31	828	210	2	637	18332	8496	39000	14460	6890	15	949
6	6	17	232	91	1	1118	49298	13856	58000	73435	32822	21	606
7	7	30	453	144	1	711	18899	7492	46000	42304	4629	14	507
8	8	35	81	49	1	1680	49346	12700	60000	347016	39737	26	120
9	9	3	11	9	1	1260	45390	13371	76000	371060	37777	30	82
10	10	34	67	41	1	4789	50665	12830	74000	319168	30132	25	225

Fact Table

After data loaded successfully into Final Project database we will deploy the cube using SQL Server Analysis (SSAS). Where we first select data source, data source views and we will select the Dimensions and Fact Table which makes the cube. We will define necessary relations and hierarchy of the dimensions and then we will process the cube and then we will run the cube which makes the data ready for the business analysis.



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

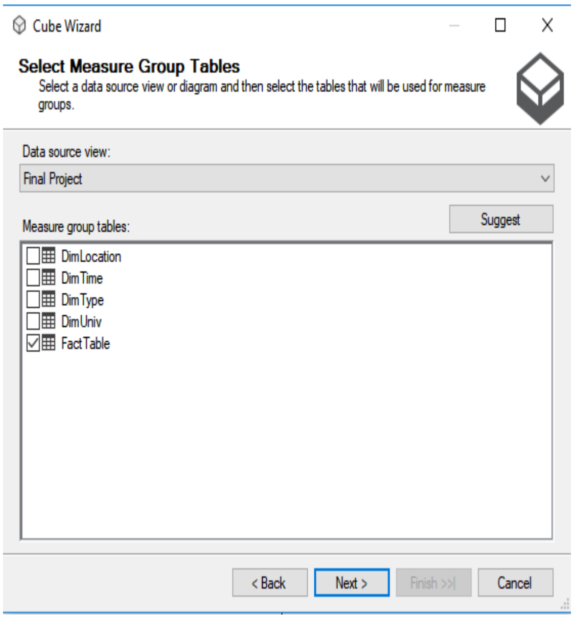
Name	Type
CwurRawData (dbo)	Table
ForbesRawData (dbo)	Table
GssRawData (dbo)	Table
RandDRawData (dbo)	Table
RawForbesData (dbo)	Table
SentimentRawData (dbo)	Table
TimesRawData (dbo)	Table
TotalRawData (dbo)	Table
WeboRawData (dbo)	Table

Included objects:

Name	Type
DimLocation (dbo)	Table
DimTime (dbo)	Table
DimType (dbo)	Table
DimUniv (dbo)	Table
FactTable (dbo)	Table

Filter:

□ Show system objects



Cube Wizard

Select Measure Group Tables

Select a data source view or diagram and then select the tables that will be used for measure groups.

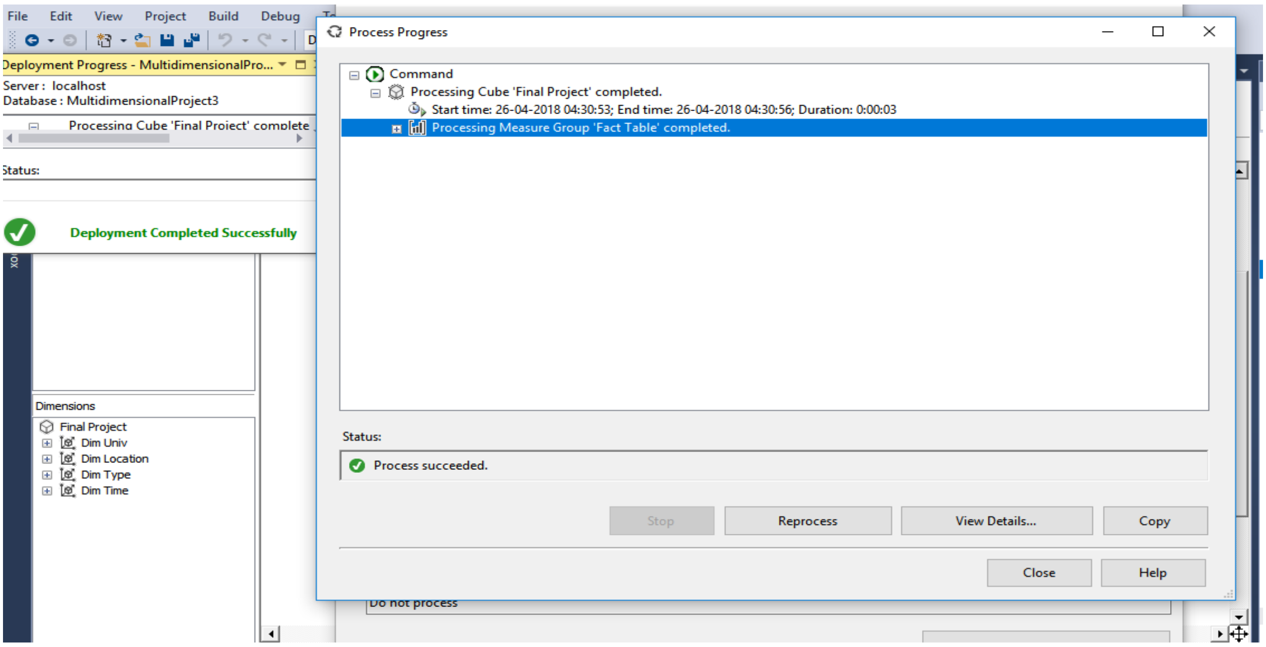
Data source view: Final Project

Measure group tables:

□ DimLocation
□ DimTime
□ DimType
□ DimUniv
☑ FactTable

Selecting tables for data view

Selecting Fact Table for cube deploy



Deployment Progress - MultidimensionalPro...

Server : localhost
Database : MultidimensionalProject3

Processing Cube 'Final Project' complete

Status:

Deployment Completed Successfully

Dimensions
Final Project
Dim Univ
Dim Location
Dim Type
Dim Time

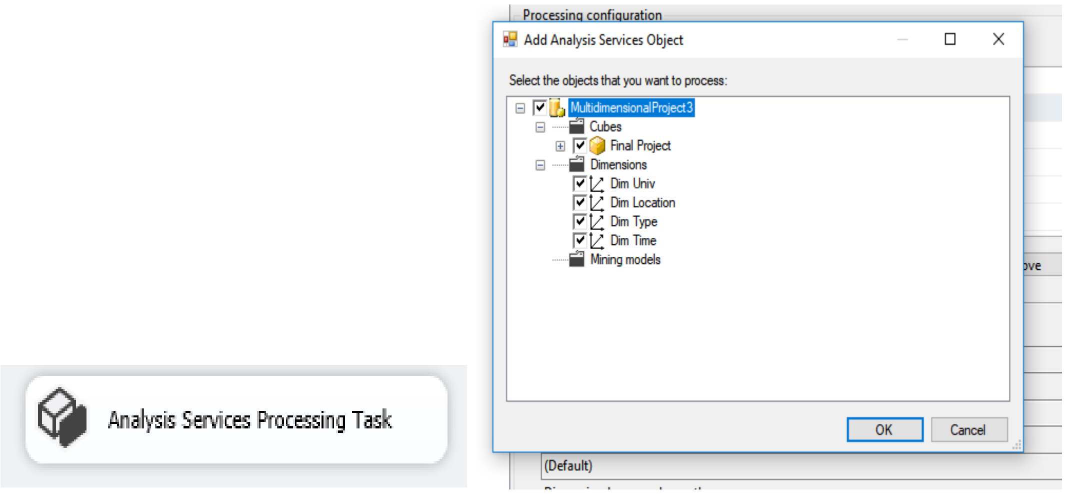
Process Progress

Command
Processing Cube 'Final Project' completed.
Start time: 26-04-2018 04:30:53; End time: 26-04-2018 04:30:56; Duration: 0:00:03
Processing Measure Group 'Fact Table' completed.

Status:
Process succeeded.

Cube Deployment and Processing

To automate the cube deployment, we will add the Analysis Services Processing Task in SSIS workflow so that it will deploy the cube automatically every time when the data in the data warehouse got updated.



Analysis Services Processing Task

Processing configuration

Add Analysis Services Object

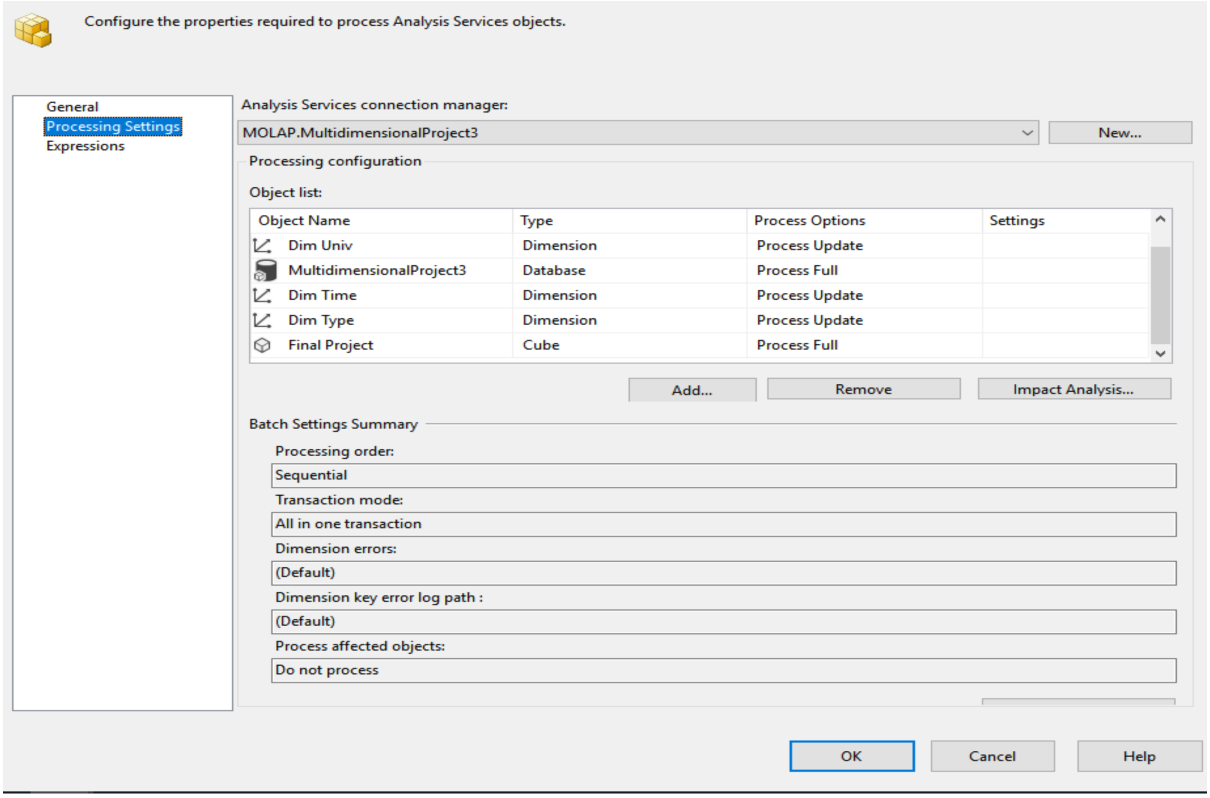
Select the objects that you want to process:

- ☑ MultidimensionalProject3
  - Cubes
    - ☑ Final Project
  - Dimensions
    - ☑ Dim Univ
    - ☑ Dim Location
    - ☑ Dim Type
    - ☑ Dim Time
  - Mining models

OK Cancel

(Default)

Analysis Services Processing Task and adding cube to analysing



Configure the properties required to process Analysis Services objects.

General
Processing Settings
Expressions

Analysis Services connection manager:
MOLAP.MultidimensionalProject3 New...

Processing configuration

Object list:

Object Name	Type	Process Options	Settings
Dim Univ	Dimension	Process Update	
MultidimensionalProject3	Database	Process Full	
Dim Time	Dimension	Process Update	
Dim Type	Dimension	Process Update	
Final Project	Cube	Process Full	

Add... Remove Impact Analysis...

Batch Settings Summary

Processing order:
Sequential

Transaction mode:
All in one transaction

Dimension errors:
(Default)

Dimension key error log path :
(Default)

Process affected objects:
Do not process

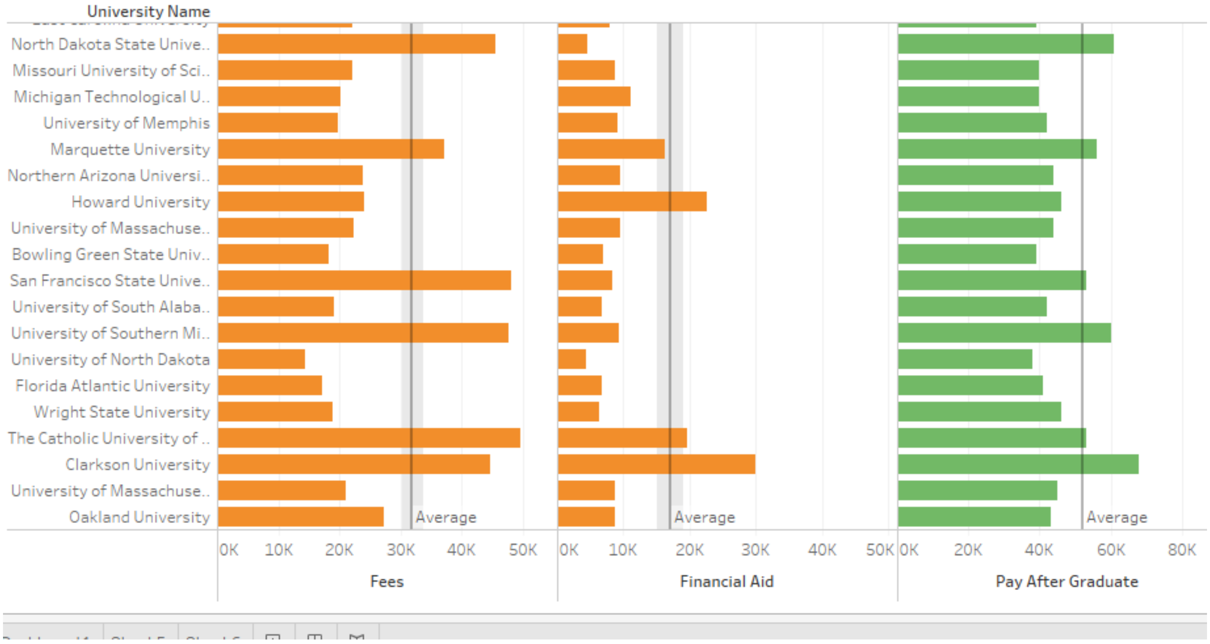
OK Cancel Help

Configuring properties of cube for automation

Application of Data Warehouse:

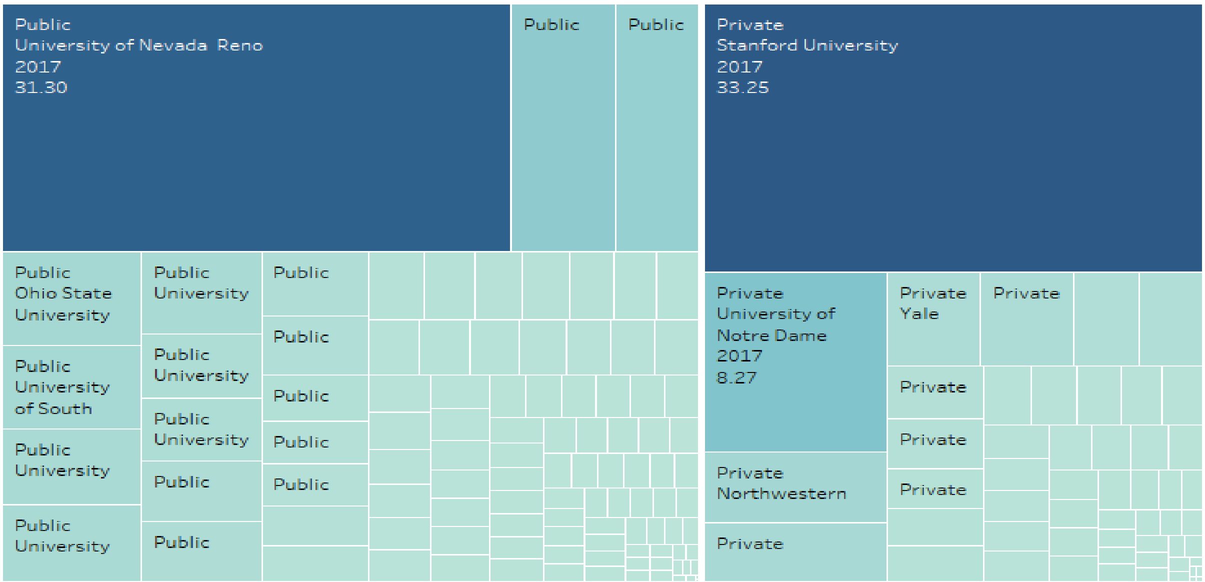
After the cube deployed successfully, now the data in the data warehouse is ready for reporting. I used Tableau which is easy and powerful visualisation tool, In the tableau I am using SQL Analysis Services to load the cube which we have deployed in previously by giving necessary credentials. My business queries are as follows

1) What is the average return of investment of student after graduation in USA?



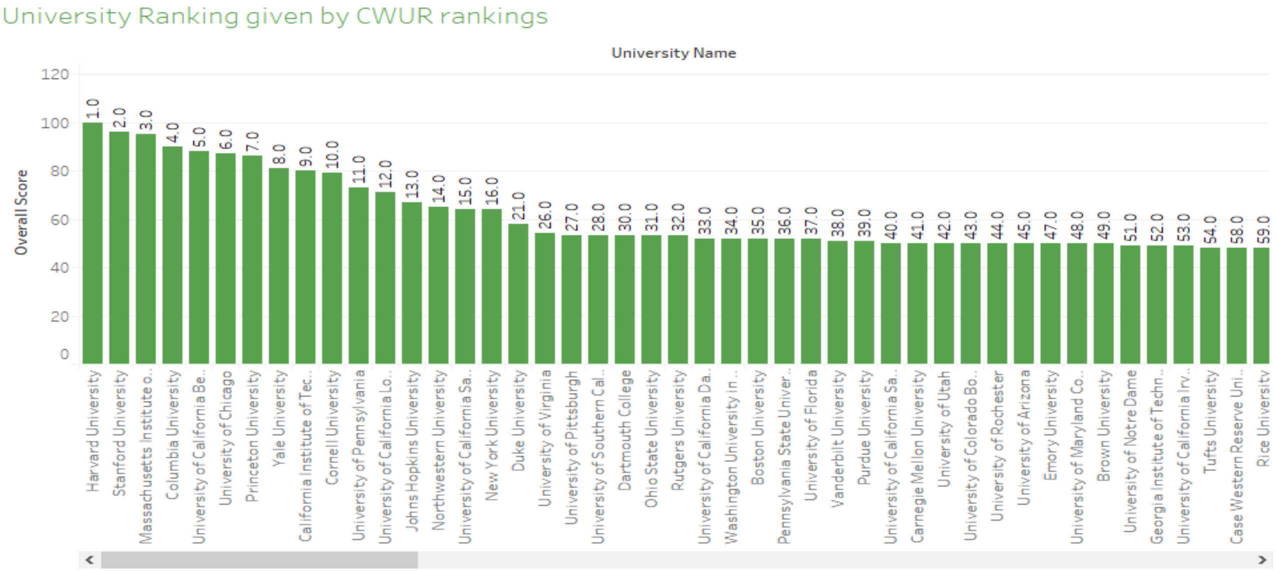
From this analysis we can observe that the average fee the student is investing, that is for fees and financial aid is 49000$, and then after the student's graduation, the average salary would be 51000$. That means there is 1% profit for completing the graduation. So, we can conclude that the student to be gaining profits, they must complete their graduation, which in-turn helps in repaying their loan if they had one.

2) What are the best universities for research-oriented graduation?

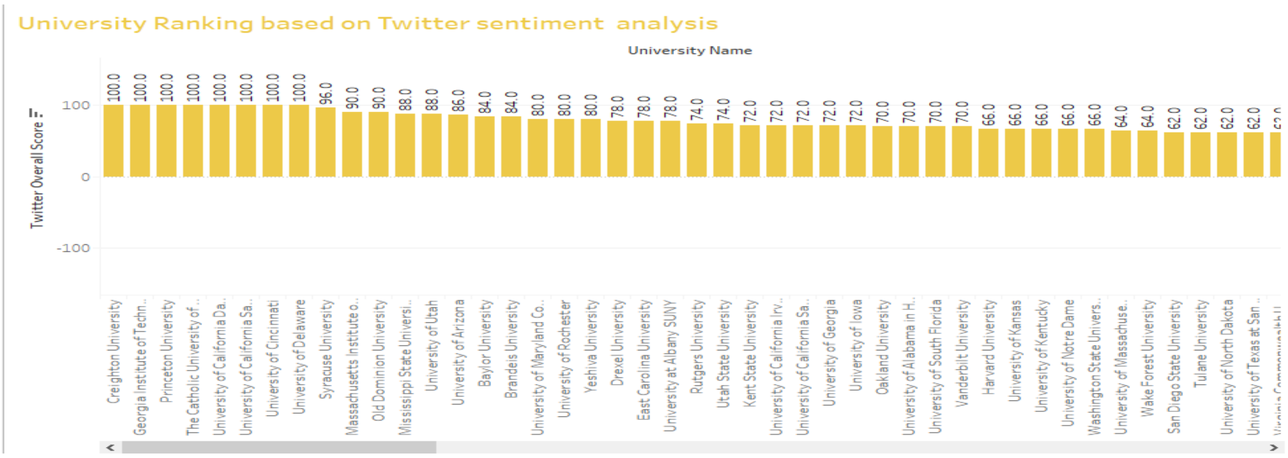


We are analysing the amount of R&D allocation to each student, we can observe that both the private and public universities are spending almost equal amount for the students to be successful. The universities like Stanford University, Notre Dame are top private universities and University of Nevada Reno, University of Akron are top public universities are investing more amount for the students in their research education.

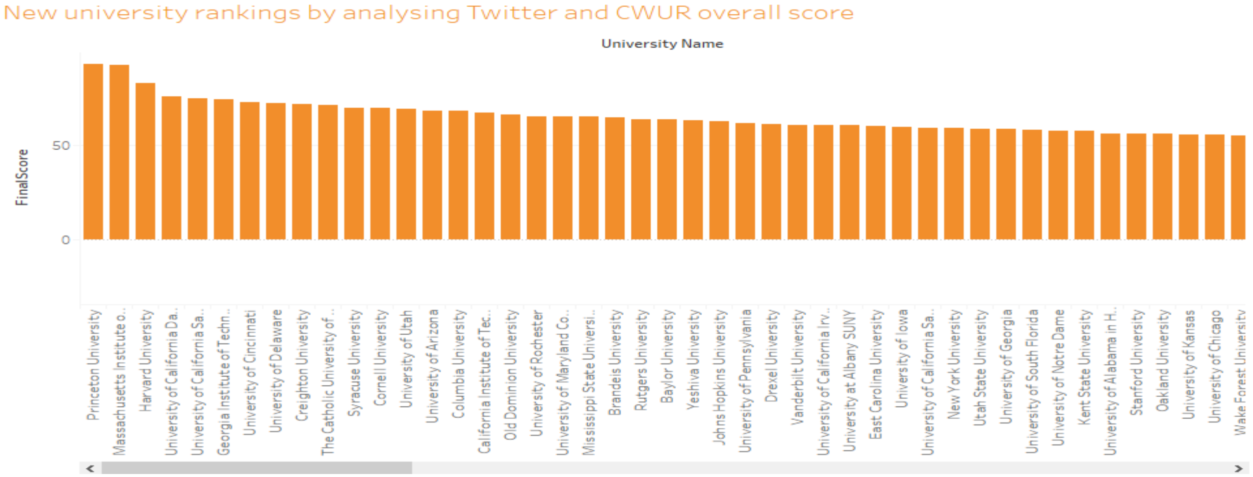
3)What are the top ranked universities when normalised ranking system is combined with rankings obtained from latest trends



(a).Rankings by CWUR overall score



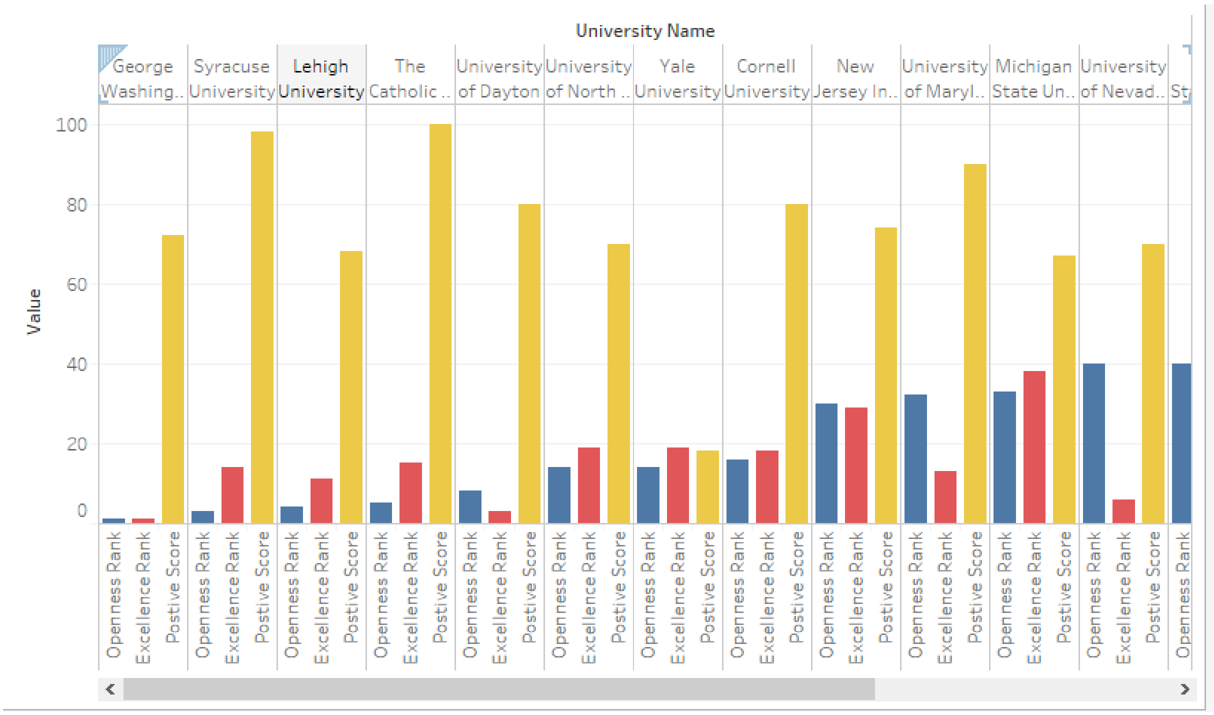
(b) Rankings by twitter score



(c) New rankings by combing the CWUR and Twitter scores

Modified university rankings by integrating CWUR (Figure a) and latest Twitter Rankings (Figure b) reveal considerable changes. this specifies latest positives or negatives affects university statures (Figure C).

4)What top universities are in Excellence and Openness for the students?



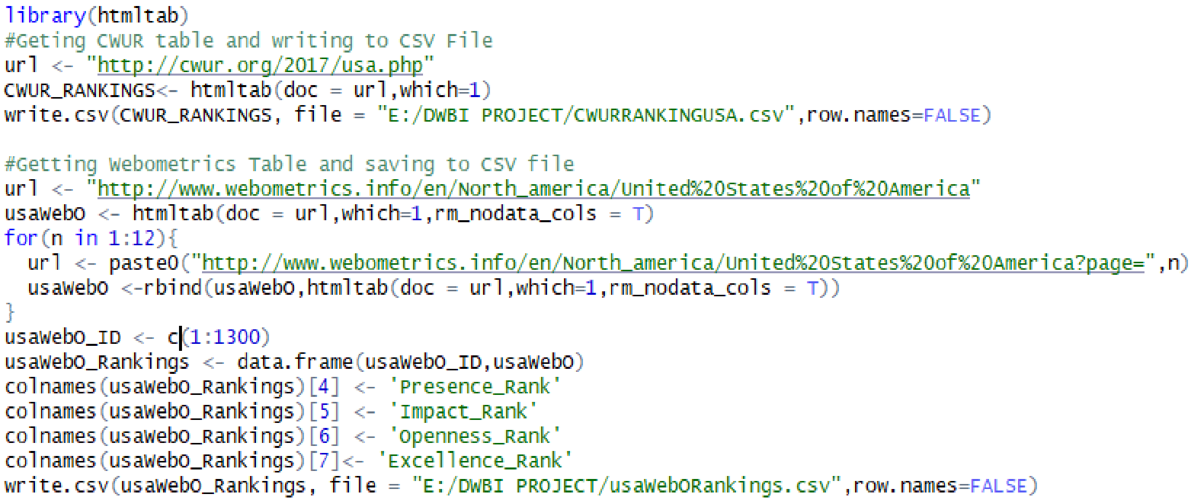
Here I am analysing whether the openness ranking which has given by webometrics is truly reflecting the people opinion in twitter. Here we can observe the universities are having the openness ranking are having positive score. Through that we can conclude that the universities are open, and they are helping students for the career growth.

References:

1) Kimball, R. and Caserta, J., 2011. The Data Warehouse? ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. John Wiley & Sons.
2) Inmon, B., 2006. DW 2.0; Architecture for the Next Generation of Data Warehousing. Information Management.
3) https://developer.twitter.com/en/docs
4) https://data-miner.io/persona-pages/persona-table

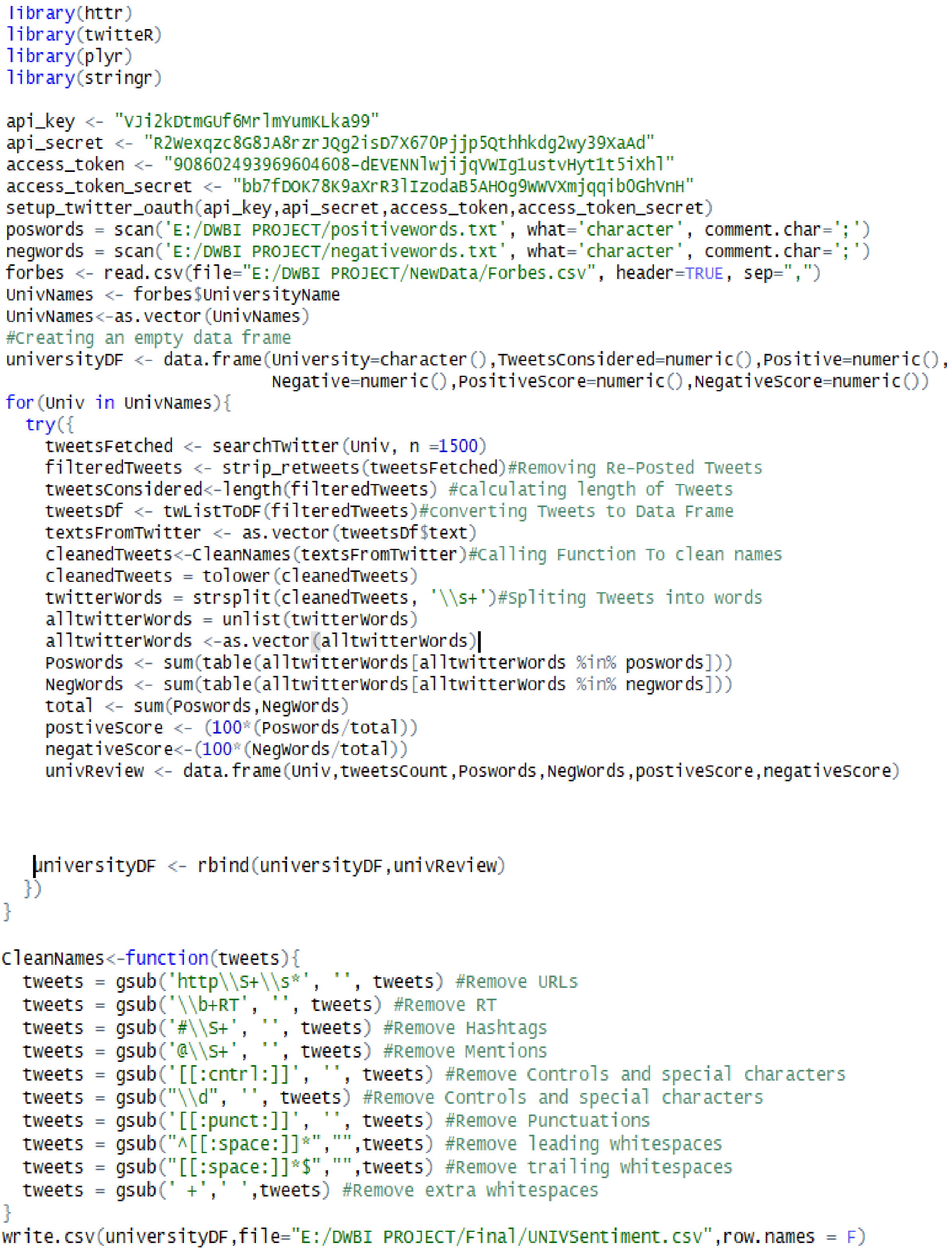
Appendix:

A) Code which is used for web scraping for CWUR, Webometrics website.



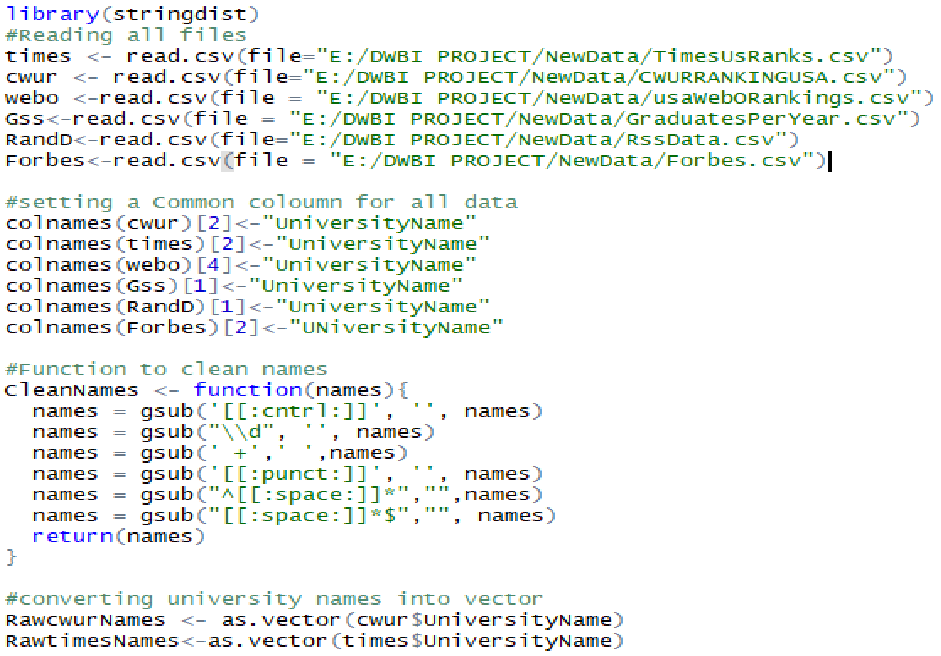
```
library(htmltab)
#Geting CWUR table and writing to CSV File
url <- "http://cwur.org/2017/usa.php"
CWUR_RANKINGS<- htmltab(doc = url,which=1)
write.csv(CWUR_RANKINGS, file = "E:/DWBI PROJECT/CWURRANKINGUSA.csv",row.names=FALSE)

#Getting Webometrics Table and saving to csv file
url <- "http://www.webometrics.info/en/North_america/United%20States%20of%20America"
usaWebo <- htmltab(doc = url,which=1,rm_nodata_cols = T)
for(n in 1:12){
  url <- paste0("http://www.webometrics.info/en/North_america/United%20States%20of%20America?page=",n)
  usaWebo <-rbind(usaWebo,htmltab(doc = url,which=1,rm_nodata_cols = T))
}
usaWebo_ID <- c(1:1300)
usaWebo_Rankings <- data.frame(usaWebo_ID,usaWebo)
colnames(usaWebo_Rankings)[4] <- 'Presence_Rank'
colnames(usaWebo_Rankings)[5] <- 'Impact_Rank'
colnames(usaWebo_Rankings)[6] <- 'Openness_Rank'
colnames(usaWebo_Rankings)[7]<- 'Excellence_Rank'
write.csv(usaWebo_Rankings, file = "E:/DWBI PROJECT/usaWebORankings.csv",row.names=FALSE)
```

B) Twitter code which is used to fetch tweets for each university

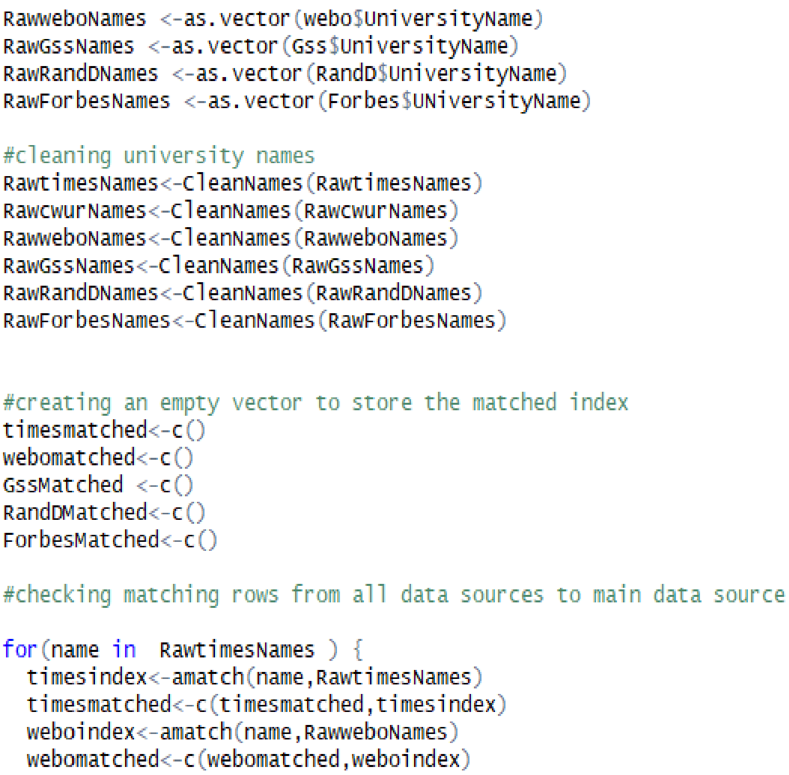


```
library(httr)
library(twitteR)
library(plyr)
library(stringr)

api_key <- "VJi2kDtmGUf6MrlmYumKLka99"
api_secret <- "R2Wexqzc8G8JA8rzrJQg2isD7X67OPjjp5Qthhkdg2wy39XaAd"
access_token <- "908602493969604608-dEVENNlwjijqVWIg1ustvHyt1t5iXhl"
access_token_secret <- "bb7fDOK78K9aXrR3lIzodaB5AHOg9WWVXmjqqibOGhVnH"
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
poswords = scan('E:/DWBI PROJECT/positivewords.txt', what='character', comment.char=';')
negwords = scan('E:/DWBI PROJECT/negativewords.txt', what='character', comment.char=';')
forbes <- read.csv(file="E:/DWBI PROJECT/NewData/Forbes.csv", header=TRUE, sep=",")
UnivNames <- forbes$UniversityName
UnivNames<-as.vector(UnivNames)
#Creating an empty data frame
universityDF <- data.frame(University=character(),TweetsConsidered=numeric(),Positive=numeric(),
                           Negative=numeric(),PositiveScore=numeric(),NegativeScore=numeric())
for(Univ in UnivNames){
  try({
    tweetsFetched <- searchTwitter(Univ, n =1500)
    filteredTweets <- strip_retweets(tweetsFetched)#Removing Re-Posted Tweets
    tweetsConsidered<-length(filteredTweets) #calculating length of Tweets
    tweetsDf <- twListToDF(filteredTweets)#converting Tweets to Data Frame
    textsFromTwitter <- as.vector(tweetsDf$text)
    cleanedTweets<-CleanNames(textsFromTwitter)#Calling Function To clean names
    cleanedTweets = tolower(cleanedTweets)
    twitterWords = strsplit(cleanedTweets, '\\s+')#Spliting Tweets into words
    alltwitterWords = unlist(twitterWords)
    alltwitterWords <-as.vector(alltwitterWords)
    Poswords <- sum(table(alltwitterWords[alltwitterWords %in% poswords]))
    NegWords <- sum(table(alltwitterWords[alltwitterWords %in% negwords]))
    total <- sum(Poswords,NegWords)
    postiveScore <- (100*(Poswords/total))
    negativeScore<-(100*(NegWords/total))
    univReview <- data.frame(Univ,tweetsCount,Poswords,NegWords,postiveScore,negativeScore)



   universityDF <- rbind(universityDF,univReview)
  })
}

CleanNames<-function(tweets){
  tweets = gsub('http\\S+\\s*', '', tweets) #Remove URLs
  tweets = gsub('\\b+RT', '', tweets) #Remove RT
  tweets = gsub('#\\S+', '', tweets) #Remove Hashtags
  tweets = gsub('@\\S+', '', tweets) #Remove Mentions
  tweets = gsub('[[:cntrl:]]', '', tweets) #Remove Controls and special characters
  tweets = gsub("\\d", '', tweets) #Remove Controls and special characters
  tweets = gsub('[[:punct:]]', '', tweets) #Remove Punctuations
  tweets = gsub("^[[:space:]]*","",tweets) #Remove leading whitespaces
  tweets = gsub("[[:space:]]*$","",tweets) #Remove trailing whitespaces
  tweets = gsub(' +',' ',tweets) #Remove extra whitespaces
}
write.csv(universityDF,file="E:/DWBI PROJECT/Final/UNIVSentiment.csv",row.names = F)
```

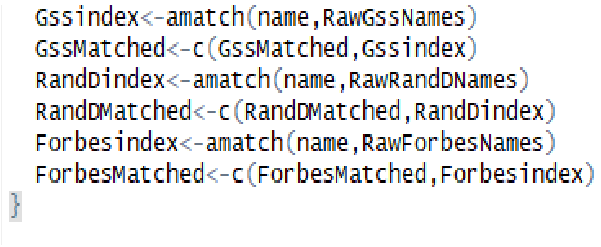
C) Code for extracting matching rows in all data sets



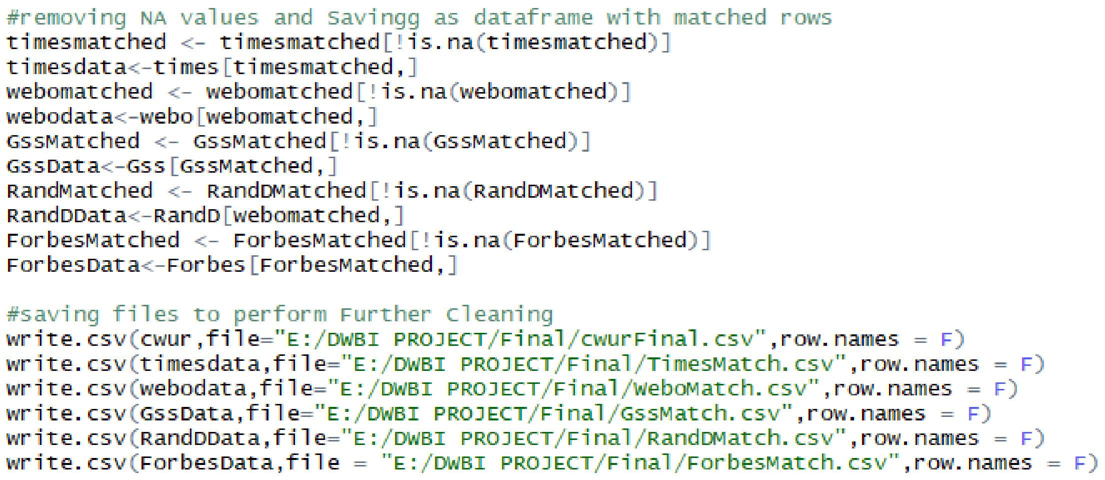
```
library(stringdist)
#Reading all files
times <- read.csv(file="E:/DWBI PROJECT/NewData/TimesUsRanks.csv")
cwur <- read.csv(file="E:/DWBI PROJECT/NewData/CWURRANKINGUSA.csv")
webo <-read.csv(file = "E:/DWBI PROJECT/NewData/usaweboRankings.csv")
Gss<-read.csv(file = "E:/DWBI PROJECT/NewData/GraduatesPerYear.csv")
RandD<-read.csv(file="E:/DWBI PROJECT/NewData/RssData.csv")
Forbes<-read.csv(file = "E:/DWBI PROJECT/NewData/Forbes.csv")

#setting a Common coloumn for all data
colnames(cwur)[2]<-"UniversityName"
colnames(times)[2]<-"UniversityName"
colnames(webo)[4]<-"UniversityName"
colnames(Gss)[1]<-"UniversityName"
colnames(RandD)[1]<-"UniversityName"
colnames(Forbes)[2]<-"UNiversityName"

#Function to clean names
CleanNames <- function(names){
  names = gsub('[[:cntrl:]]', '', names)
  names = gsub("\\d", '', names)
  names = gsub(' +',' ',names)
  names = gsub('[[:punct:]]', '', names)
  names = gsub("^[[:space:]]*","",names)
  names = gsub("[[:space:]]*$","", names)
  return(names)
}

#converting university names into vector
RawcwurNames <- as.vector(cwur$UniversityName)
RawtimesNames<-as.vector(times$UniversityName)
```



```
RawweboNames <-as.vector(webo$UniversityName)
RawGssNames <-as.vector(Gss$UniversityName)
RawRandDNames <-as.vector(RandD$UniversityName)
RawForbesNames <-as.vector(Forbes$UNiversityName)

#cleaning university names
RawtimesNames<-CleanNames(RawtimesNames)
RawcwurNames<-CleanNames(RawcwurNames)
RawweboNames<-CleanNames(RawweboNames)
RawGssNames<-CleanNames(RawGssNames)
RawRandDNames<-CleanNames(RawRandDNames)
RawForbesNames<-CleanNames(RawForbesNames)


#creating an empty vector to store the matched index
timesmatched<-c()
webomatched<-c()
GssMatched <-c()
RandDMatched<-c()
ForbesMatched<-c()

#checking matching rows from all data sources to main data source

for(name in  RawtimesNames ) {
  timesindex<-amatch(name,RawtimesNames)
  timesmatched<-c(timesmatched,timesindex)
  weboindex<-amatch(name,RawweboNames)
  webomatched<-c(webomatched,weboindex)
```



```
  Gssindex<-amatch(name,RawGssNames)
  GssMatched<-c(GssMatched,Gssindex)
  RandDindex<-amatch(name,RawRandDNames)
  RandDMatched<-c(RandDMatched,RandDindex)
  Forbesindex<-amatch(name,RawForbesNames)
  ForbesMatched<-c(ForbesMatched,Forbesindex)
}
```



```r
#removing NA values and Savingg as dataframe with matched rows
timesmatched <- timesmatched[!is.na(timesmatched)]
timesdata<-times[timesmatched,]
webomatched <- webomatched[!is.na(webomatched)]
webodata<-webo[webomatched,]
GssMatched <- GssMatched[!is.na(GssMatched)]
GssData<-Gss[GssMatched,]
RandMatched <- RandDMatched[!is.na(RandDMatched)]
RandDData<-RandD[webomatched,]
ForbesMatched <- ForbesMatched[!is.na(ForbesMatched)]
ForbesData<-Forbes[ForbesMatched,]

#saving files to perform Further Cleaning
write.csv(cwur,file="E:/DWBI PROJECT/Final/cwurFinal.csv",row.names = F)
write.csv(timesdata,file="E:/DWBI PROJECT/Final/TimesMatch.csv",row.names = F)
write.csv(webodata,file="E:/DWBI PROJECT/Final/WeboMatch.csv",row.names = F)
write.csv(GssData,file="E:/DWBI PROJECT/Final/GssMatch.csv",row.names = F)
write.csv(RandDData,file="E:/DWBI PROJECT/Final/RandDMatch.csv",row.names = F)
write.csv(ForbesData,file = "E:/DWBI PROJECT/Final/ForbesMatch.csv",row.names = F)
```