
Data Warehousing and Business Intelligence
Mohammed Abou Hassan

INTRODUCTION:

The objective behind this work is to create a fully functional data warehouse for an imaginary Statistics company called “StatsIE” that gathers and analyses information about internet service providers in Ireland. StatsIE will fetch the largest amount of information possible about different Telecommunication companies operating in Ireland from different sources and analyse them to have a good insight of the Irish market. StatsIE needs to build a data warehouse so it can study the present Irish market and make future predictions.

DATA SOURCES:

StatsIE has obtained data about market share for each company operating in Ireland by quarters from 2015 until 2017 from www.comreg.ie (Commission for Communications Regulation). StatsIE will also fetch data from Twitter to perform a sentiment analysis about tweets from people living in Ireland about their internet providers. Furthermore StatsIE will scrape the forum of the Irish website www.ratemyisp.ie to extract the star rating and more importantly to perform a sentiment analysis on the text and paragraphs written by people describing their experience with their providers (Was good opportunity to try the skills I learned from CA1 about Sentiment Analysis).

Also StatsIE decided on fetching data about best internet speeds around the country. www.Speedtest.net (OOKLAH) has data about max speeds atteined around the Ireland by operator which made the website a good source of data for StatsIE.

The max speeds fetched from the website are labelled “Max speeds” that will be seen in one the columns in the Data warehouse. Extra details about company phone number, email address and address were generated in Mockaroo and fetched with “Mockaroo API schema” using R so the whole process stays automated (only secondary data were fetched from Mockaroo that are not going to be used in the fact table, just to keep the numbers and the study real).

ARCHITECTURE:

Kimball’s method was used to build StatsIE data warehouse as it provides many advantages over other methods, some of these advantages are:

- It requires less time to be built, which was needed here considering we have a limited time to finish our project.
- While StatsIE is a big enterprise, it wanted to build a data warehouse that focuses on the Irish internet sector which why Kimball’s way of building a warehouse was followed. Kimball’s focuses on a specific business process/team and not the whole enterprise [1].
- Unlike other models (Inmon’s in particular) Kimball’s data warehouse needs less people to maintain which means a lower cost and this is perfect in the case of StatsIE [2][3].
- StatsIE used Kimball’s “Star Schema” which is a multidimensional design where dimensions are arranged around a fact table. Denormalization of tables was followed as well because it is useful in dealing with the proliferation of this schema [4]. Denormalization will provide better performance and a smarter data structure for

data warehouse users to navigate. A real example followed by StatsIE about denormalization: The Market shares data that StatsIE found included 4 quarters from 2015 to 2017. StatsIE had to deformalize the table by having one column for year, one column for share, another for quarter and a last one for Operator (redundant data will appear in the last two mentioned).

6 major steps were followed to build StatsIE Data Warehouse:

1- Extracting data from the sources mentioned before, using R scripts in SSIS
2- Storing the data on the disk which is the start of the staging phase
3- Cleaning data using R
4- Transforming data according to business objectives and organizing them in a tabular form
5- Loading data in dimensions and fact tables using SSIS giving them keys
6- Testing tables and checking for integrity
7- Deploying and processing the cube

DATA WAREHOUSE DATA MODEL:

Data that were fetched from the sources are stored in staging tables in OLEDB in preparation to be inserted into the dimension tables. All staging and dimensional tables were created using SQL task editors in SSIS which also populated them in later steps.

Each one of the sources will lead to the creation of one staging table. So, in total StatsIE made 4 staging tables and called them: Stagequarters, ISP_Staging, Tweet_Staging and Staging_Com.

Twitter will provide sentiment score extracted from tweets, ratemyisp.ie will provide sentiment score extracted from the forum along with a star rating for each company, comreg.ie will provide data for the staging table about the market share per quarter per company while Speedtest.net will provide maximum speeds.

Three additional secondary rows were created in Mockaroo (Email, Address and Phone number) and imported using API Schema with R.

As mentioned in the previous section, StatsIE followed Kimball's star schema in its data warehouse building. The dimensions used are:

1- DimDate: StatsIE fetched data about the market share from an official Irish website and these data state the shares per quarter per year. So, in our date dimension the two attributes used are quarter and year. R was used here to fetch the data and transform the excel sheet to a data frame after a series of transformations including denormalization. StatsIE used year and quarter in this dimension to use them later in any analysis. For example if they want to perform any drill down, they can go from a yearly to a quarterly level. Hierarchy being: Year -> Quarter

2- Dim_company: It will include company details like name, address, email, phone number and most importantly max internet speed attained. The reason we have max speed here is because it will be a good idea to see if the speed affects people's opinions about the service.

3- Dim_Cust_Opinion: This dimension has scores from Twitter sentiment analysis, scores from ratemyisp sentiment analysis and ratemyisp star rating. This dimension is an important one as it provides the fact table with a good chunk of its values. This dimension will give us a good insight of what people think about a certain company.

4- Dim_Market: This dimension has the market share beside the operator ID. Data that were fetched from www.comreg.ie and staged is to be inserted in this dimension table. Combining the market share with the date dimension will give us a good insight when compared with scores and ratings as well as when compared with speeds.

It is worth mentioning all our dimension tables have been automatically assigned keys in SSIS and that is after creating the key column in SSMS using the following Syntax:

```
[****_Key] INT IDENTITY(1,1) NOT NULL PRIMARY KEY
```

ETL

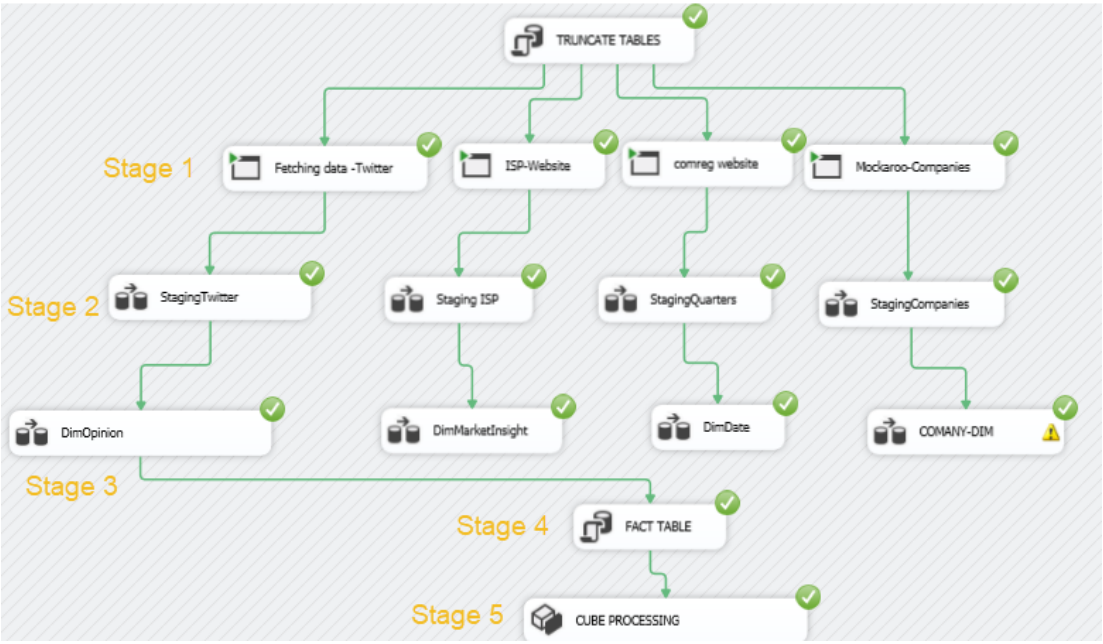
ETL means extraction, transformation and loading. StatsIE automated the process from the point of extraction of data until the cube is processed.

The screenshot below was taken from StatsIE data warehouse and it illustrates the flow of data from the source to the cube.

SSIS integration in Visual Studio was chosen to serve as an ETL. It offers running R script inside task editors which makes fetching data from the internet and transforming them a seamless task.

The screenshot below was taken to illustrate the automated process:

(The first SQL editor called Truncate tables is there to delete all data from tables preparing them to receive fresh data again).



Stage 1:

This is where fetching data starts. R scripts were included in “Execute Process” that run inside visual studio. (Please note all the R scripts used in this project will be present at the end of this document)

a- The first one to the left fetches unstructured data from Twitter using an API. In this script mentions and hashtags about operators were used to fetch tweets of people talking about their experience with providers. Mentions like and hashtags like@vodafone and #vodafone were used to capture the sentiments which are later to be transformed in the same script to a table of scores, positive negative and total twitter score. The same script will take care of any dirty data that appear along the way like emoticons or characters with a different encoding that confuse R and lead to the production of a dirty table. In StatsIE case, most of the encoding problems faced in R were solved by just adding: sky.text <- sapply(,function(row) iconv(row, "latin1", "ASCII", sub=""))

b- The second one labelled ISP-Website targets the ratemyisp.ie website and its forum to fetch cutomer’s feelings about an operator. Star rating was also captured along the way. (XPATH was used here to fetch specific objects of the page and not all of it)

StatsIE used their expertise with sentiment analysis to create a script that scans people’s emotions expressed in paragraphs on the forum and give a final score. At the end of this process StatsIE will have an isp.csv file that has isp scores and isp star rating. The same file will be considered as an automatic input to stage 2.

c- The third “Execute process” labelled comreg website will target the official Irish website and specifically a file that contains data about Irish market shares. It is a file in the form of an excel sheet that has details about market shares from 2015 until 2017 for the Irish telecom industry. XLConnet package was used in R to download the file, choose the right columns and rows and discard the rest. Then change the data needed into a data frame and save it on an quarter.csv file. It is worth mentioning here StatsIE found it a bit difficult to make this package work with their computers as this package depends on windows Java. XLConnect is one of the few packages in R that have external dependencies. After solving Java problem, StatsIE stored quarter.csv file locally to be used later for automatic loading of staging tables.

d- The fourth “Mockaroo-companies” Execute Process connects to Mockaroo API schema that will fetch pre-prepared columns including operators name, address, email address and phone number. It also connects to speedtest.net to fetch a table having max speeds scored by different operators in Ireland.

Same as a, b and c, .csv file will be written and used automatically to load staging tables.

Stage 2:

All the .csv files that were stored locally, will be used by SSIS to create staging tables.
Before inserting the files from .csv to the staging tables, staging tables should be created, and SQL Task Editors are used for this process.

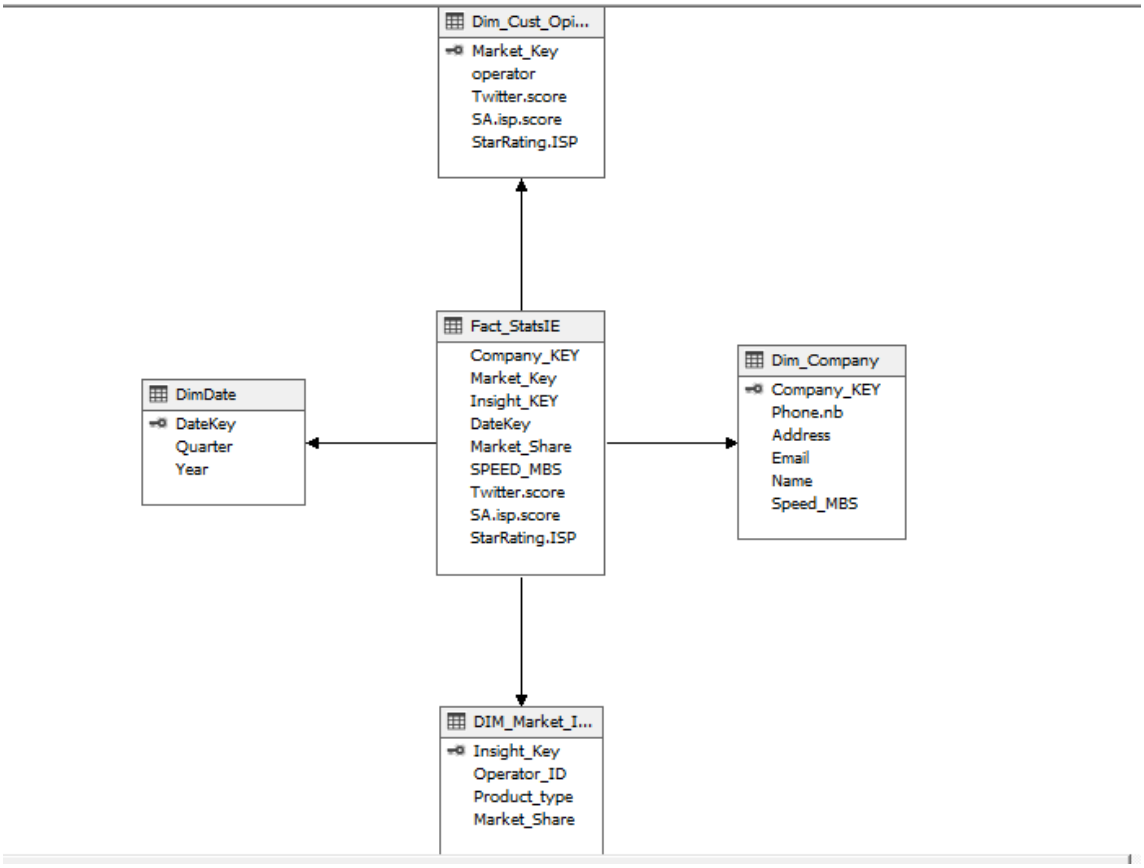
After creating the empty staging tables, StatsIE needs to populate them. 4 Data flows were created for this purpose where each of the 4 has a couple of tasks: Flat file source and OLE DB destination.
The flat file source is the output of stage 1 and the OLEDB destination is the staging table. Data type were re-adjusted when uploading the .csv flat file source as SSIS did not detect them properly.

Stage3:

This is the stage where StatsIE uploads the data to dimensions. Empty dimension tables are created first with SQL queries and then SSIS populates them from staging tables. Data flow tasks are used again but in this stage each data flow task has OLEDB source and OLEDB destination.
In this stage multiple staging tables to fill one dimension or it could be the case of 1:1. StatsIE combined Twitter staging table and ISP staging table into one dimension called Dim_Cust_Opinion.

Furthermore, the “Quarters” staging table was used to feed the DimDate and Dim_Market_Insight.
DimDate has quarter and year only as a lone dimension (following Kimball’s star schema).

Stage 4:



Knowing that all our dimension tables are created and populated with proper automated data flows, it is time to build our fact table.
As a first step, an empty table was created with all the facts we need to include, these are all the numerical values that are important to all analysis processes.

The fact also includes foreign keys that refer to each of the dimension tables.

The screenshot above is an illustration of the schema used in StatsIE data warehouse.

Stage 5:
This stage is for cube deployment. SSIS offers Analysis Services Processing Task that can run and process a cube. It can browse the cube too if needed.
It is a quick and handy way to make sure the whole process including the cube is fully automated.

At this stage StatsIE's warehouse will populate with a push of a button.

All processes in the warehouse are automated and the Truncate table at the top makes sure no data are being duplicated.

APLICATION OF DATA WAREHOUSE:

Business Queries:

Query 1: “Sources are Twitter and comreg.ie”

In this query StatsIE is going to use data from two different data sources, Twitter and www.comreg.ie.
As discussed before, from Twitter we are going to receive a final score while comreg.ie will provide us with market shares. It is important to note that market shares statistics provide by this website provide data for only 4 companies that share around 90% of the Irish market (depending on the quarter and the year).

The 10% left is shared between small Irish broadband providers (27 of them exactly) with no exact numbers released about them. They were not individually included in any official statistics (cso.ie, …).
StatsIE is mostly interested in studying the statistically significant internet providers which means this study is going to focus on the biggest 4 companies.

The companies were coded: Eir = 1, Virgin Media = 2, Vodafone = 3, Digiweb = 4.

Using Tableau and connecting it to the cube, StatsIE was able to see how the market share and people’s opinion about a certain company are related.

Is there any relation between market shares and people’s opinions?

To answer the above question, StatsIE used three variables in Tableau, market shares, company and total twitter score. StatsIR could see an interesting pattern.
Eir with a market share around 35% scored 70/100 in Twitter, Virging Media with a market shares around 28% scored 61% and Vodafone with market share of 17% scored 33% on twitter, Digiweb 100% Twitter score.

StatsIE decided to consider the Digiweb’s score as an outlier and that is because Digiweb had a total of 6 tweets which is much less than 30 tweets to be statistically significant.

	Twitter score	Market Shares
Eir	70%	35%
Virgin Media	61%	28%
Vodafone	33%	17%

(The rest of the companies had a total number of tweets bigger than 30 which why they will be included in this study).

The pattern is clear here, the more positivity is present among customers towards an operator the more market shares the company is going to have.

People start subscribing more when the company is offering a satisfying service.

Query 2: "Sources are ratemyisp.ie and Twitter"

In this business query, SatsIE is going to use two different sources:
Twitter and ratemyisp.ie
Again, and same as the query before it, we are going to be studying the 4 biggest companies that share around 90% of market shares. (excluding the one that scored 100% in Twitter with 6 tweets)

Using Tableau and connecting to our famous cube, StatsIE wanted to answer an important question for their business:

To add a public feeling about each of the company they are profiling, what system is more reliable to use? Star rating on the ratemyisp website or Twitter score?

The results given by Tableau showed some interesting results:

	Twitter score	ISP Score
Eir	70%	22%
Virgin Media	61%	52%
Vodafone	33%	33%

Comparing Twitter sentiment Analysis to the sentiment Analysis of the forum we can see that in the case of Virgin Media and Vodafone the two score are very close.

While in the case of Eir, the company was owned by Eircom before and ratemyisp wesite has merged the old and the new comments in one place labelled Eir (formely Eircom). Which means Eir has inherited some of the bad reviews and feeling from old comments and reviews.

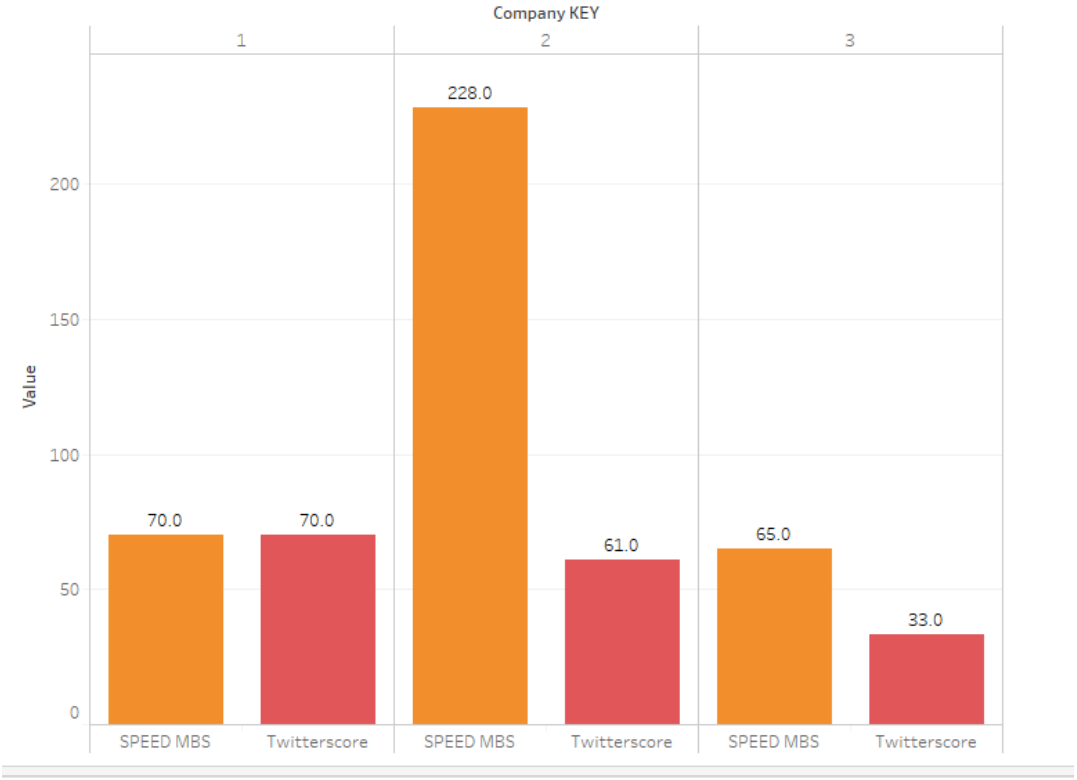
So as an answer to the question posed by StatsIE about what system to follow, StatsIE wanted to do more testing. Star rating was introduced to check if team who build the system that scrapes and analyses the forum is trustworthy.

Star rating from ratemyisp was included to the analysis (out of 5), and the results came out in favour of the website.
1, 2 and 2 out of 5 (in order) means Eir is 20%, 40% for Virgin and 40 % for Vodafone.

The results are very close to ISP scores which proves the system used to scrap the forum is trustworthy if SatstIE decided to use it in the future.

Query 3:" The two sources are Twitter and Speedtest.net."

StatsIE wants to know if speed of the network has anything to do with market share? With people's feelings towards an operator?



Charts in Tableau are easy to make and very informative.

From one look we can conclude speeds are not the most important factor when it comes to internet services.
Virgin Media which is the second on the list has a max speed of 228 MB/S which is much higher than Vodafone and Eir.
But its Twitter score is not the best among the other companies as we can see from the chart above.

On the other hand, Vodafone's max speed is very close to Eir's one, but its Twitter's score is much less.

Depending on these results and as a conclusion, StatsIE decides that max speed is not a pointer for good service, and decided this model of studying these two variables alone is not a good model and needs more factors to be added like consistency of speeds maybe (speed ususally goes down during peak times) or price!

Query 4:"Sources are ratemyisp.ie and comreg.ie"

StatsIE wants to answer a final question:

Are customers opinions expressed on the forum of ratemyisp.ie related to the number of shares (from comreg.ie) each company has?

The analysis was done like the first three queries and the results illustrated in Tableau were the following:

	ISP score	Market Shares
Virgin Media	52%	28%
Vodafone	33%	17%
Digiweb	30%	2%

Eir to be excluded from this study because ratemyisp.ie combined Eir and Eircom together on the forum.

According to these numbers, market share decreases with isp score. Having said that, Vodafone and Digiweb ISP scores were too close to be considered as a good model.

Conclusion:

This work tried to draw conclusions about the Irish Telecom market by looking at different elements. All results are directly related to the real present Irish market.
It is worth noting StatsIE tried to get additional financial details about the Irish telecom companies like profit, revenue, Churn, etc but it was unsuccessful as these details are not usually disclosed by private companies.
As a future improvement to this data warehouse StatsIE will add more elements to the date warehouse like demographics, use by gender and maybe weather depending on the data that will be available for public.

Please note R scrips are present at the end of the document, one for Twitter sentiment analysis, a second for ratemyisp sentiment analysis and extraction of star rating, a third for fetching and transforming excel sheet from comreg.ie and a fourth for fetching speeds from speedtest.net and companies secondary details from Mockaroo API.

REFERENCES

[1] Kimball, R. and Ross, M. (2013) *The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling*, *Vasa*. doi: 10.1145/945721.945741.

[2] Chaudhuri, S. and Dayal, U. (1997) 'An overview of data warehousing and OLAP technology', *ACM SIGMOD Record*, 26(1), pp. 65–74. doi: 10.1145/248603.248616.

[3] Lawyer, J. and Chowdhury, S. (2004) 'Best practices in data warehousing to support business initiatives and needs', in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, p. 9 pp. doi: 10.1109/HICSS.2004.1265515.

[4] Shin, S. K. and Sanders, G. L. (2006) 'Denormalization strategies for data retrieval from data warehouses', *Decision Support Systems*, 42(1), pp. 267–282. doi: 10.1016/j.dss.2004.12.004.

R scripts were created with the help of Stackoverflow

R SCRIPTS

COMREG.IE

```
###  Downloads the Excel sheet for the source nd loads it into R ###
setwd("C:/Users/MOLAP/Documents/R")
options(java.home="C:\\Program Files\\Java\\jre1.8.0_151")
library(XLConnect)
url = "http://www.comreg.ie/media/2017/11/5.-Internet-Data-Q2-2017.xlsx"
file = tempfile()
download.file(url,file, mode ='wb')
quarters <-readWorksheetFromFile(file = file, sheet = 1, header = FALSE, startCol = 42, startRow = 79, endCol = 51, endRow = 85)
unlink(file)

quarters$group = cumsum(quarters[, 1]==1)
colnames(quarters) <- c("2015 Q1", "2015 Q2", "2015 Q3", "2015 Q4", "2016 Q1", "2016 Q2", "2016 Q3", "2016 Q4", "2017 Q1", "2017 Q2", "Operator_ID")
quarters <- quarters [-1,]
rownames(quarters) <- c(1:6)
quarters$Operator_ID <- c('1','2', '3', '4', '5', '6')
quarters$$peeds_MBS <- c('70.32', '228.18', '65.90', 'NULL', '60.21', 'NULL')
df <- data.frame(Res = c(quarters, recursive = TRUE))



df$Year <- c("2015", "2015", "2015", "2015", "2015", "2015", "2015", "2015", "2015", "2015", "2015", "2015","2015", "2015", "2015", "2015", "2015","2015", "2015", "2015", "2015",
"2015", "2015", "2016", "2016", "2016", "2016", "2016", "2016","2016", "2016", "2016", "2016", "2016", "2016","2016", "2016", "2016", "2016", "2016", "2016",
"2016", "2016","2016", "2016", "2016", "2016", "2016", "2016","2016", "2016", "2016", "2016", "2016", "2016", "2017", "2017", "2017", "2017", "2017", "2017","2017", "2017", "2017", "2017",
"2017", "2017")
df$Quarter <- c("1", "1", "1", "1", "1", "1", "2", "2", "2", "2", "2", "2", "3", "3", "3", "3", "3", "3", "4", "4", "4", "4", "4", "4")

df$Operator_ID <- c("Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO", "Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky",
"OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir",
"Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO","Eir", "Virgin Media",
"Vodafone", "Digiweb", "Sky", "OAO", "Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO", "Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky", "OAO")
colnames(df)[1]<- "Quarter"
#colnames(df$year) <-"year"
colnames(df)[1] <- "Market Share"
df <- df[-(61:72), ]

df[] <- lapply(df, sub, pattern = "%", replacement = "")
df[is.na(df)] <- 0
write.csv(df, 'quarters.csv', row.names = FALSE)
df
```

TWITTER

```
library(dplyr)
library(devtools)
library(htmltab)
#install_github("geoffjentry/twitteR")
library(twitteR)
library(plyr)
library(doBy)
library(gtools)
library(rvest)
api_key<- "rZarKou6K8xncDzD8JQCH6xeA"
api_secret <- "da9xXJ3AdUVpaniSp6bAScPEpnJWfh2mFuDNk5rVubOsKn7uyb"
access_token <- "371460606-S7NTyGKV6DsOIr1aRhWN5yOfiKWjiIw15UzE2Zit"
access_token_secret <- "vVeTMkggAhTgqXlUqKMytfDOXvMxASa1GO0zMGUbaHX98"

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
sky.tweets = searchTwitter('@skyireland', n=1500)
virgin.tweets = searchTwitter('@virginmediaie', n=1500)
digiweb.tweets = searchTwitter('@digiweb_ireland', n=1500)
eir.tweets = searchTwitter('@Eir', n=1500)
vodafone.tweets = searchTwitter('@vodafoneireland', n=1500)

hu.liu.pos = scan('C:/Users/MOLAP/Documents/positive-words.txt', what='character', comment.char=';')
hu.liu.neg = scan('C:/Users/MOLAP/Documents/negative-words.txt', what='character', comment.char=';')
pos.words = c(hu.liu.pos)
neg.words = c(hu.liu.neg)


score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{
require(plyr)
require(stringr)

    scores = laply(sentences, function(sentence, pos.words, neg.words) {
   sentence = gsub('[[:punct:]]', '', sentence)
   sentence = gsub('[[:cntrl:]]', '', sentence)
   sentence = gsub('\\d+', '',sentence)
   sentence = tolower(sentence)
   word.list = str_split(sentence, '\\s+')
   words = unlist(word.list)
   pos.matches = match(words, pos.words)
   neg.matches = match(words, neg.words)
   pos.matches = !is.na(pos.matches)
   neg.matches = !is.na(neg.matches)
   score = sum(pos.matches) - sum(neg.matches)
   return(score)
  }, pos.words, neg.words, .progress=.progress )
  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
sky.text = laply(sky.tweets, function(t) t$getText())
sky.text <- sapply(sky.text,function(row) iconv(row, "latin1", "ASCII", sub=""))
```

```
digiweb.text = laply(digiweb.tweets, function(t) t$getText())
digiweb.text <- sapply(digiweb.text,function(row) iconv(row, "latin1", "ASCII", sub=""))
vodafone.text = laply(vodafone.tweets, function(t) t$getText())
vodafone.text <- sapply(vodafone.text,function(row) iconv(row, "latin1", "ASCII", sub=""))
virgin.text = laply(virgin.tweets, function(t) t$getText())
virgin.text <- sapply(virgin.text,function(row) iconv(row, "latin1", "ASCII", sub=""))
eir.text = laply(eir.tweets, function(t) t$getText())
eir.text <- sapply(eir.text,function(row) iconv(row, "latin1", "ASCII", sub=""))

virgin.score = score.sentiment(virgin.text, pos.words, neg.words, .progress='text')
sky.score = score.sentiment(sky.text, pos.words, neg.words, .progress='text')
digiweb.score = score.sentiment(digiweb.text, pos.words, neg.words, .progress='text')
eir.score = score.sentiment(eir.text, pos.words, neg.words, .progress='text')
vodafone.score = score.sentiment(vodafone.text, pos.words, neg.words, .progress='text')

library(RCurl)
digiweb.score$operator = 'digiweb'
sky.score$operator = 'sky'
eir.score$operator = 'Eir'
virgin.score$operator = 'virgin'
vodafone.score$operator = 'vodafone'

all.scores = smartbind(virgin.score,digiweb.score, sky.score, eir.score, vodafone.score)
all.scores$very.pos = as.numeric(all.scores$score >= 2 )
all.scores$very.neg = as.numeric(all.scores$score <= -2)
##' for each airline(airline+code) lets use the ratio of very positive to very negative tweets as the overall sentiment score of each company
twitter.df = ddply(all.scores, c('operator'), summarise, pos.count = sum(very.pos), neg.count = sum( very.neg ) )
twitter.df$all.count = twitter.df$pos.count + twitter.df$neg.count
twitter.df$Twitter.score = round( 100 * twitter.df$pos.count/twitter.df$all.count )
twitter.df <- twitter.df[order(twitter.df$operator, decreasing = TRUE),]

twitter.df[] <- lapply(twitter.df, sub, pattern = "NaN", replacement = "0")
write.csv(twitter.df, file = "twitter.csv", row.names = FALSE)
```

RATEMYISP.IE

```
library(dplyr)
library(devtools)
library(htmltab)

library(twitteR)
library(plyr)
library(doBy)
library(gtools)
library(rvest)
setwd("C:/Users/MOLAP/Documents/R")



hu.liu.pos = scan('C:/Users/MOLAP/Documents/positive-words.txt', what='character', comment.char=';')
hu.liu.neg = scan('C:/Users/MOLAP/Documents/negative-words.txt', what='character', comment.char=';')
pos.words = c(hu.liu.pos)
neg.words = c(hu.liu.neg)


score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{
 require(plyr)
 require(stringr)

 scores = laply(sentences, function(sentence, pos.words, neg.words) {
  sentence = gsub('[[:punct:]]', '', sentence)
  sentence = gsub('[[:cntrl:]]', '', sentence)
  sentence = gsub('\\d+', '',sentence)
  sentence = tolower(sentence)
  word.list = str_split(sentence, '\\s+')
  words = unlist(word.list)
  pos.matches = match(words, pos.words)
  neg.matches = match(words, neg.words)
  pos.matches = !is.na(pos.matches)
  neg.matches = !is.na(neg.matches)
  score = sum(pos.matches) - sum(neg.matches)
  return(score)
 }, pos.words, neg.words, .progress=.progress )
 scores.df = data.frame(score=scores, text=sentences)
 return(scores.df)
}

library(XML)
dat <- readLines("http://ratemyisp.ie/ratings/", warn=FALSE)
raw2 <- htmlTreeParse(dat, useInternalNodes = TRUE)
dat2 <- readLines("http://ratemyisp.ie/ratings/page/2/", warn=FALSE)
raw3 <- htmlTreeParse(dat2, useInternalNodes = TRUE)



Result.virgin <- xpathSApply(raw2,"//div[7]/div[3]/ul/li[1]/span/img[1]", xmlGetAttr, "alt")
virgin.df <- do.call("rbind", lapply(Result.virgin, as.data.frame))
virgin.df$operator = c('virgin')
names(virgin.df) <- c("Rating.ISP", "operator")

Result.sky <- xpathSApply(raw2,"//div[14]/div[3]/ul/li[1]/span/img[1]", xmlGetAttr, "alt")
sky.df <- do.call("rbind", lapply(Result.sky, as.data.frame))
sky.df$operator = c('sky')
names(sky.df) <- c("Rating.ISP", "operator")
#des.sky <- xpathSApply(pagetree, "//div[14]/div[2]/text()", xmlValue)

Result.digiweb <- xpathSApply(raw2,"//div[20]/div[3]/ul/li[1]/span/img[1]", xmlGetAttr, "alt")
```

```
digiweb.df <- do.call("rbind", lapply(Result.digiweb, as.data.frame))
digiweb.df$operator = c('digiweb')
names(digiweb.df) <- c("Rating.ISP", "operator")
#des.digiweb <- xpathSApply(pagetree, "//div[20]/div[2]/text()", xmlValue)

Result.eir <- xpathSApply(raw3,"//div[5]/div[3]/ul/li[1]/span/img[1]", xmlGetAttr, "alt")
eir.df <- do.call("rbind", lapply(Result.eir, as.data.frame))
eir.df$operator = c('eir')
names(eir.df) <- c("Rating.ISP", "operator")

Result.vodafone <- xpathSApply(raw3,"//div[9]/div[3]/ul/li[1]/span/img[1]", xmlGetAttr, "alt")
vodafone.df <- do.call("rbind", lapply(Result.vodafone, as.data.frame))
vodafone.df$operator = c('vodafone')
names(vodafone.df) <- c("Rating.ISP", "operator")


total.isp = rbind(vodafone.df, eir.df, sky.df, digiweb.df, virgin.df)

url <- "http://ratemyisp.ie/ratings/vodafone/"
vodafone <- read_html(url)
vodafone <- vodafone%>%html_nodes(".wrapper")%>%html_text()
url <- "http://ratemyisp.ie/ratings/virginmedia/"
virgin <- read_html(url)
virgin <- virgin%>%html_nodes(".wrapper")%>%html_text()

url <- "http://ratemyisp.ie/ratings/sky-broadband/"
skybrod <- read_html(url)
skybrod <- skybrod%>%html_nodes(".wrapper")%>%html_text()
url <- "http://ratemyisp.ie/ratings/eircom/"
Eir <- read_html(url)
Eir <- Eir%>%html_nodes(".wrapper")%>%html_text()


url <- "http://ratemyisp.ie/ratings/digiweb/"
digiweb <- read_html(url)
digiweb <- digiweb%>%html_nodes(".wrapper")%>%html_text()




vodafone.isp.score = score.sentiment(vodafone, pos.words, neg.words, .progress='text')
virgin.isp.score = score.sentiment(virgin, pos.words, neg.words, .progress='text')
eir.isp.score = score.sentiment(Eir, pos.words, neg.words, .progress='text')
sky.isp.score = score.sentiment(skybrod, pos.words, neg.words, .progress='text')
digiweb.isp.score = score.sentiment(digiweb, pos.words, neg.words, .progress='text')

vodafone.isp.score$operator = 'vodafone'
digiweb.isp.score$operator = 'digiweb'
sky.isp.score$operator = 'sky'
eir.isp.score$operator = 'Eir'
virgin.isp.score$operator = 'virgin'
all.scores.isp = smartbind(virgin.isp.score, sky.isp.score, digiweb.isp.score, eir.isp.score, vodafone.isp.score)
all.scores.isp$very.pos = as.numeric(all.scores.isp$score >= 2 )
all.scores.isp$very.neg = as.numeric(all.scores.isp$score <= -2)
##' for each airline(airline+code) lets use the ratio of very positive to very negative tweets as the overall sentiment score of each company
isp.df = ddply(all.scores.isp, c('operator'), summarise, pos.count = sum(very.pos), neg.count = sum( very.neg ) )
isp.df$all.count = isp.df$pos.count + isp.df$neg.count
isp.df$score.reviews = round( 100 * isp.df$pos.count/isp.df$all.count )
colnames(isp.df)[5] <- "SA.isp.score"

x <- merge(isp.df, total.isp, by = 'operator')
x<- x[order(x$operator, decreasing = TRUE),]
write.csv(x, file = "isp.csv", row.names = FALSE)

x
```

Mockaroo and Speednet:

```
library(RJSONIO)
library(RCurl)
library(jsonlite)
# grab the data
raw_data <- getURL("https://my.api.mockaroo.com/company.json?key=18897e90")
# covert from JSON into a list in R
dataJSON <- fromJSON(raw_data)
vector1 <- dataJSON$Operator_id
vector2 <- dataJSON$Phone
vector3 <- dataJSON$Address
vector4 <- dataJSON$email
vector5 <- c("Eir", "Virgin Media", "Vodafone", "Digiweb", "Sky",  "Premier Laois", "East Cork", "Nova Broadband", "Wicklow Broadband", "Reach Broadband", "Rapid Broadband", "Premier
Broadband South", "Regional Broadband", "Pure Broadband", "Smart Broadband", "Magnet Broadband", "O2 Broadband", "Airwire", "LightNet", "Irish Broadband", "QSat", "Onwave Broadband",
"Perlico", "Meteor Broadband", "Leinster Broadband", "KBO", "Imagine" , "UTV", "WTS", "Ice Broadband", "Three", "Choice Rural Broadband", "Tesco")
Companies <- cbind(vector1, vector2, vector3, vector4, vector5)
colnames(Companies)<-c('Operator_ID', 'Phone nb', 'Address', 'Email', 'Name')
Companies <-  as.data.frame(Companies)
Companies$Speeds_MBS <- c('70.32', '228.18', '65.90', '0', '0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0')
class(Companies)
write.csv(Companies, file = "company.csv", row.names = FALSE
```