DATA WAREHOUSE AND BUSINESS INTELLIGENCE PROJECT

HOTELS

Submitted by

Anusha Madduri

X17164991@student.ncirl.ie

Msc Data Analytics

National College of Ireland

Introduction:

The growth of data has been increasing enormously day by day. Around 2.5 quintillion bytes of data is generated per day. Handling large amounts of data becomes quite difficult. Raw Data is converted into useful information which can be further utilized to take business decisions and to enhance customer experience. So, Data Warehouse and Business intelligence become essential in all the companies.

Why Hotels Data warehouse?

The main purpose of this project is building data warehouse for hotels which is one of the leading hospitality industries. On daily basis hotels around the world generate lots of data. This data can be used to derive good amount of business intelligence. The derived business intelligence can be used to improve hotel's marketing strategy and revenue management. If used in a right way, it will help hotels around the world increase profits and daily online visits and also help to gain loyal customers. In addition, the derived info can also be used to improve the digital marketing strategies for the hotel.

The project includes collection of data from different sources and finally it provides business insights, strategies to enhance customer experience, and finding the drawbacks in hotel management and architecture. This report covers the following steps:

1. Gathering of related data from different sources.

2. Automatic extraction, cleaning, transformation of data.

3. Making a design of data warehouse.

4. Extraction, transformation and loading of data into data warehouse .

5.Creating cube using data warehouse for analysis purpose.

6. Reporting business intelligence (BI) queries.

Sources of Data:

I have taken three sources of data which includes two structured datasets (MakeMyTrip and Goibibo) and one unstructured dataset (TripAdvisor). MakeMyTrip and Goibibo are different online travel companies located in India. TripAdvisor is the American company which provides reviews on hotels and restaurants.

1. MakeMyTrip : It is one of the structured datasets which I have chosen. It is in the CSV format and contains about 20000 rows. Taken from data.world

2. Goibibo : This dataset contains around 4000 hotels information in India. The format of the file is CSV. This dataset was taken from Kaggle.

3. TripAdvisor : For hotel reviews, I scrapped the hotel links on Tripadvisor and performed sentiment analysis on reviews to find the positive and negative sentiment score of each hotel. Around 300 hotel links on Tripadvisor were scrapped to get the reviews.

Data Warehouse Architecture:

In this project, to build data warehouse, Kimball approach is used which is also known as bottom up approach **[1]**. It requires star schema architecture for creating dimensional modelling. Star schema is the simplest data warehouse schema which is well suited for this project. Fact table is surrounded by dimensional tables in star schema which resembles star. In star schema, Fact table is in 3NF and dimension tables are denormalized. This project requires dimension tables which should be in denormalized form.

Kimball approach is widely supported by number of business intelligence tools **[1]**. In kimball approach, reporting requirements are tactical and business oriented. In this project, hotels data warehouse is used for business insights and enhance customer experience. Therefore, I consider Kimball approach is suitable for building hotels data warehouse.

Below is the data warehouse model for implementing hotels data warehouse. MakeMyTrip, Goibibo, Tripadvisor are the three data sources.

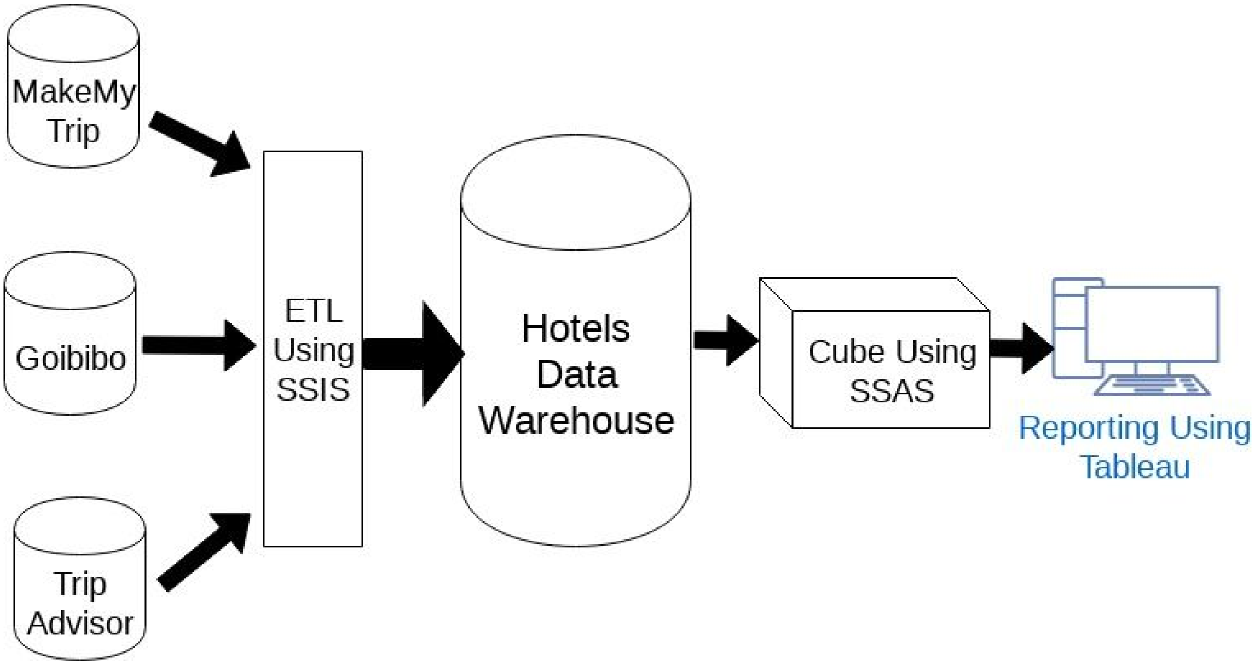


Figure 1: Data warehouse model

Data Warehouse Data Model:

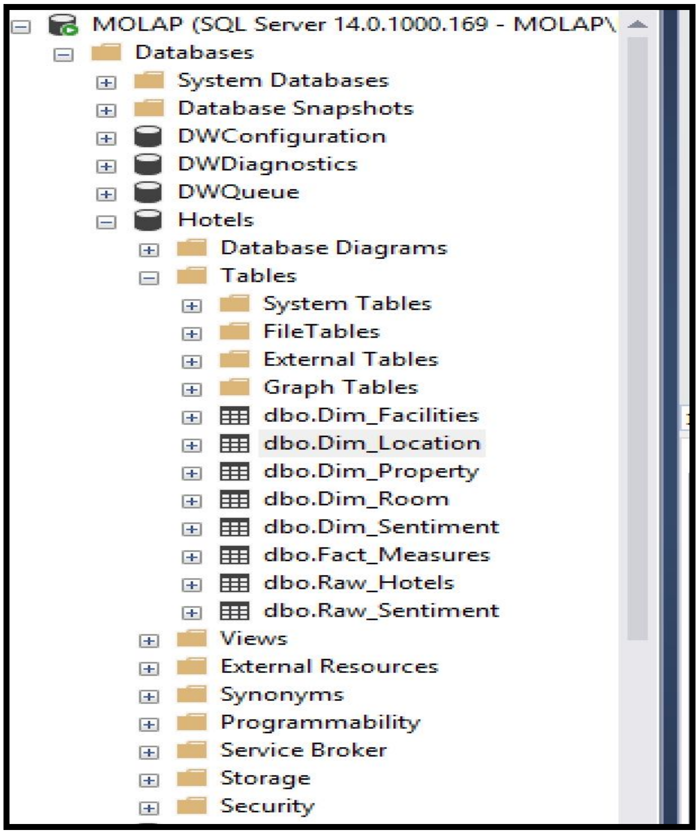
The design and implementation of data warehouse is based on bottom up approach. According to Kimball approach, the same is as follows:

- **Select the business process**: Here the business process is to study ratings and reviews of hotels based on the facilities, point of interests around hotels and location feasibility.
- **Declare the grain**: We need to identify the grain of business process. Grain identification is important for creating perfect model for dimensions. In this project, grain is the user ratings of the hotels and it helps to figure out the possible dimensions.
- **Identify the dimensions**: Dimension tables are identified and created with primary keys. Dimension attributes are generally descriptive or textual values. Dimension tables are smaller in size compared to fact table.

- **Identify the fact table**: Fact table is created with the help of dimension tables. The primary keys of dimension tables become foreign keys in fact table. In general, fact table have foreign key columns and measures.

Data Warehouse Design:

In this project, I created a database (named as hotels), dimension tables and fact table using SSMS. The created database and tables are shown below.



```
⊟ MOLAP (SQL Server 14.0.1000.169 - MOLAP\
    ⊟ Databases
        ⊞ System Databases
        ⊞ Database Snapshots
        ⊞ DWConfiguration
        ⊞ DWDiagnostics
        ⊞ DWQueue
        ⊟ Hotels
            ⊞ Database Diagrams
            ⊟ Tables
                ⊞ System Tables
                ⊞ FileTables
                ⊞ External Tables
                ⊞ Graph Tables
                ⊞ dbo.Dim_Facilities
                ⊞ dbo.Dim_Location
                ⊞ dbo.Dim_Property
                ⊞ dbo.Dim_Room
                ⊞ dbo.Dim_Sentiment
                ⊞ dbo.Fact_Measures
                ⊞ dbo.Raw_Hotels
                ⊞ dbo.Raw_Sentiment
            ⊞ Views
            ⊞ External Resources
            ⊞ Synonyms
            ⊞ Programmability
            ⊞ Service Broker
            ⊞ Storage
            ⊞ Security
```

Figure 2: Created tables in SSMS

After creating dimension and fact tables in SSMS, populate dimension and fact tables using SSIS (SQL Server Integration Services). The workflow is shown below:

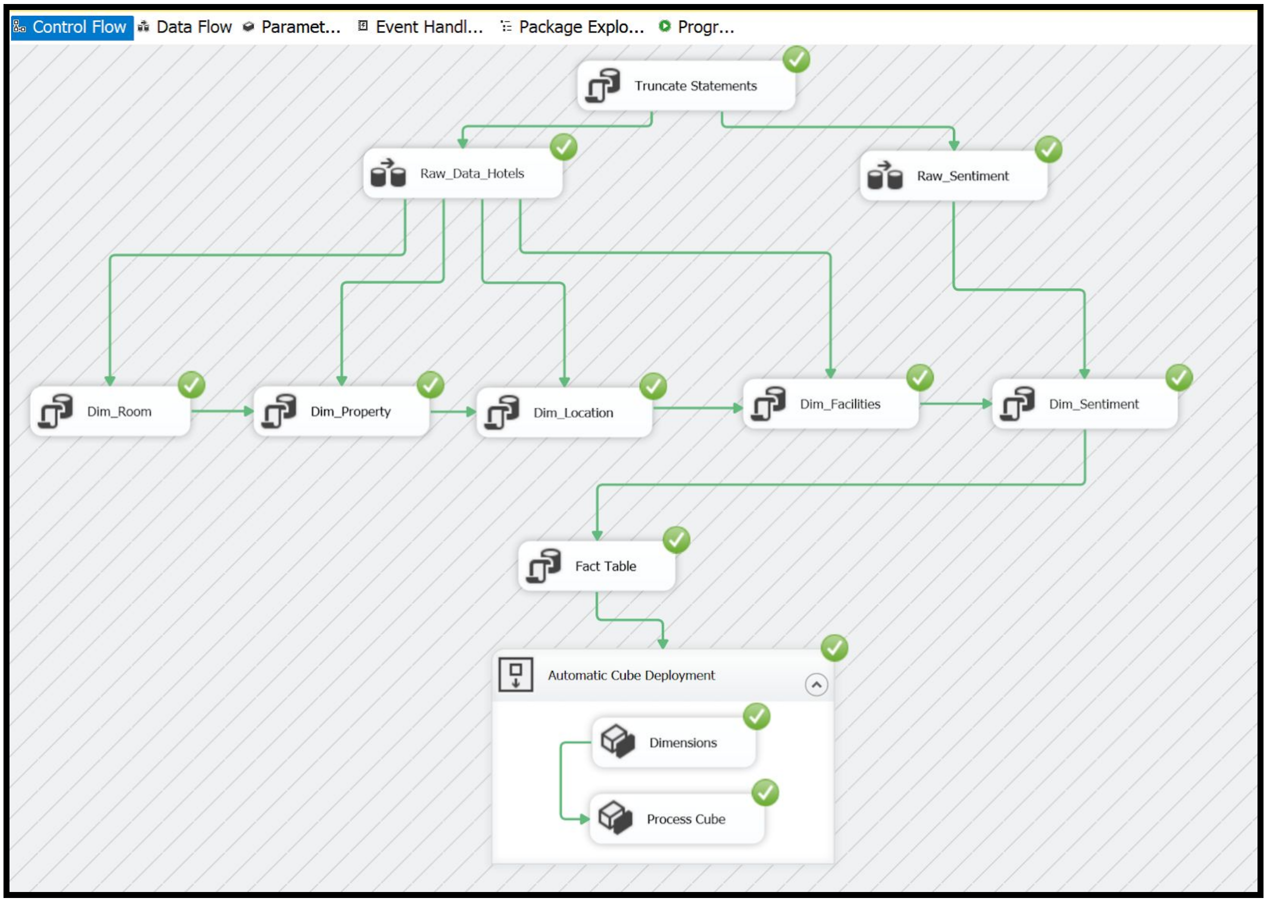
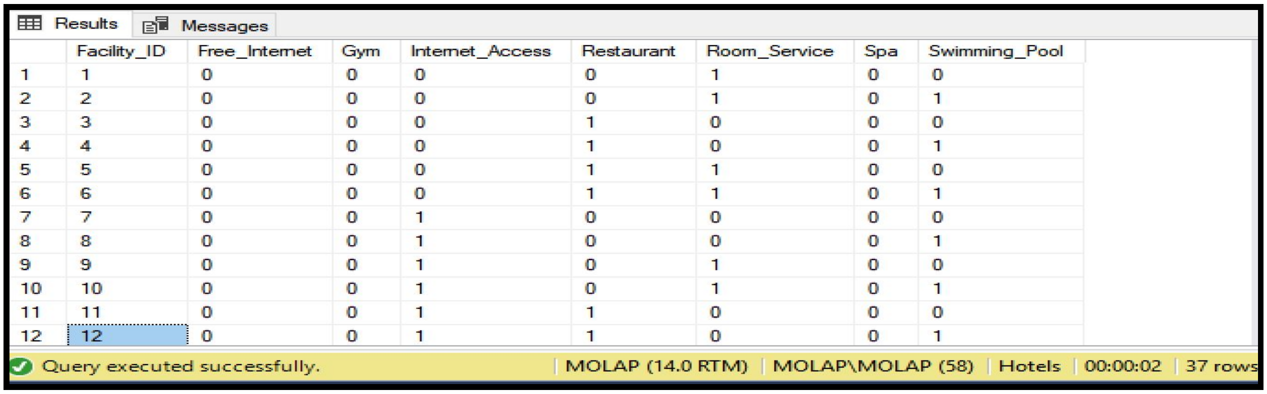


Figure 3:SSIS Workflow

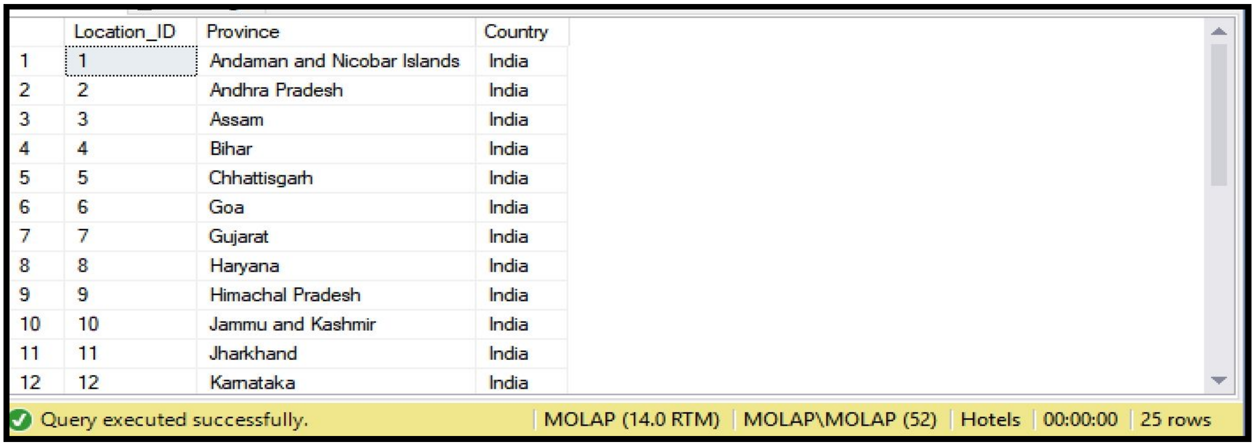
The workflow shown above with green tick marks indicates that, data in dimension and fact tables is populated successfully. Truncate statements (Execute SQL task) is used to truncate all tables before populating. If truncate command is not used during while running workflow, it tries to insert rows again and again. This leads to primary key violation. In second step, raw data is extracted using data flow task. **Raw_Data_Hotels** indicates the merged structured data sets and **Raw_Sentiment** indicates unstructured data. The third step is to populate data into dimension tables and finally fact table got updated. The screen shots of dimension tables and fact table are shown below:



	Facility_ID	Free_Internet	Gym	Internet_Access	Restaurant	Room_Service	Spa	Swimming_Pool
1	1	0	0	0	0	1	0	0
2	2	0	0	0	0	1	0	1
3	3	0	0	0	1	0	0	0
4	4	0	0	0	1	0	0	1
5	5	0	0	0	1	1	0	0
6	6	0	0	0	1	1	0	1
7	7	0	0	1	0	0	0	0
8	8	0	0	1	0	0	0	1
9	9	0	0	1	0	1	0	0
10	10	0	0	1	0	1	0	1
11	11	0	0	1	1	0	0	0
12	12	0	0	1	1	0	0	1

✓ Query executed successfully. | MOLAP (14.0 RTM) | MOLAP\MOLAP (58) | Hotels | 00:00:02 | 37 rows

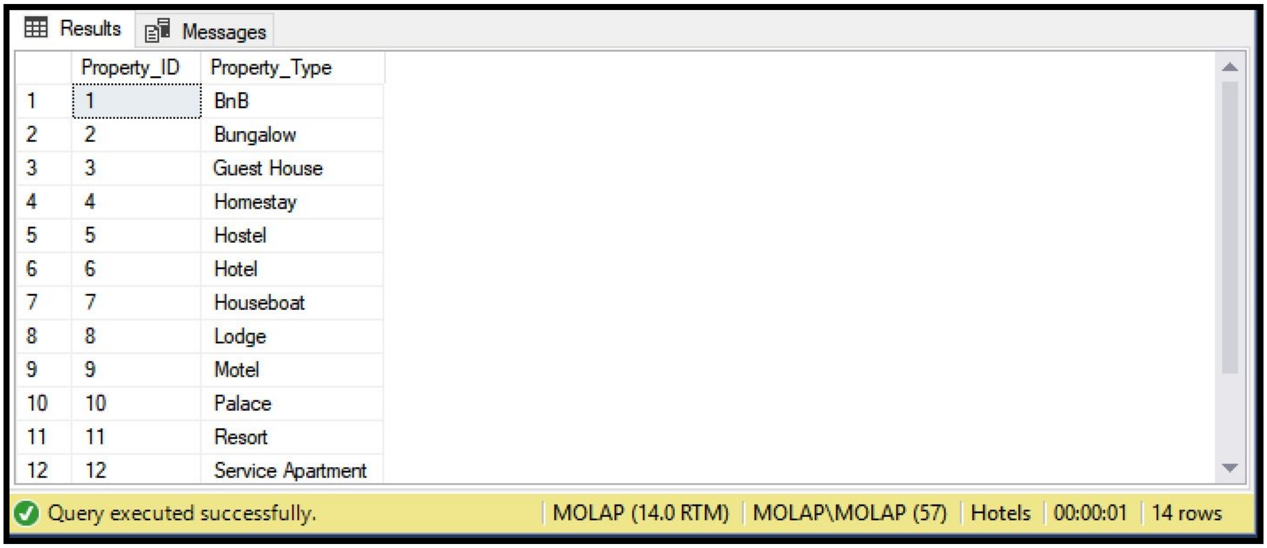
Figure 4: Facility dimension table



	Location_ID	Province	Country
1	1	Andaman and Nicobar Islands	India
2	2	Andhra Pradesh	India
3	3	Assam	India
4	4	Bihar	India
5	5	Chhattisgarh	India
6	6	Goa	India
7	7	Gujarat	India
8	8	Haryana	India
9	9	Himachal Pradesh	India
10	10	Jammu and Kashmir	India
11	11	Jharkhand	India
12	12	Karnataka	India

✓ Query executed successfully. | MOLAP (14.0 RTM) | MOLAP\MOLAP (52) | Hotels | 00:00:00 | 25 rows

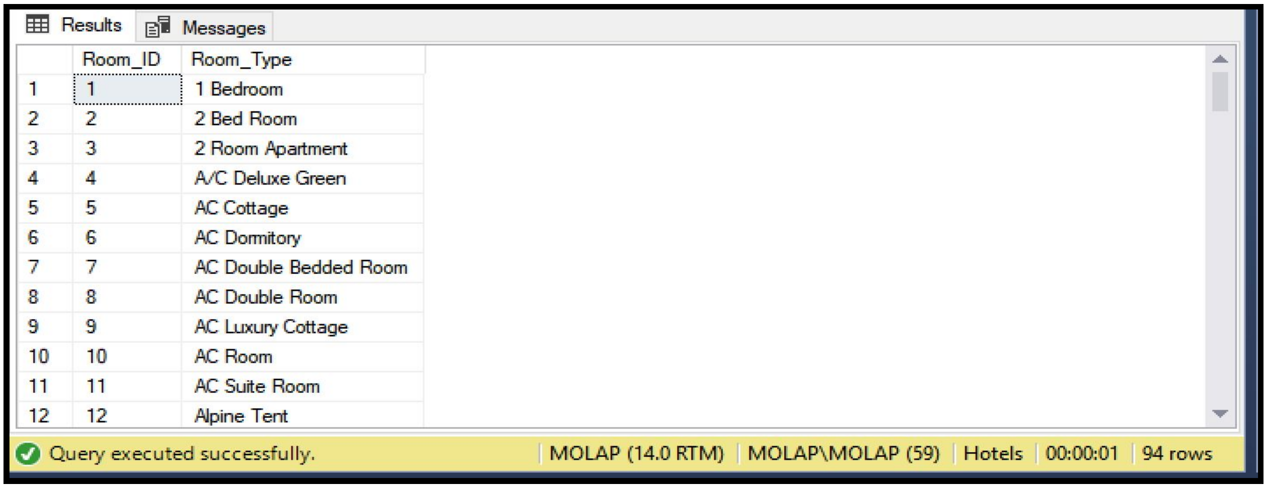
Figure 5: Location dimension table



	Property_ID	Property_Type
1	1	BnB
2	2	Bungalow
3	3	Guest House
4	4	Homestay
5	5	Hostel
6	6	Hotel
7	7	Houseboat
8	8	Lodge
9	9	Motel
10	10	Palace
11	11	Resort
12	12	Service Apartment

✓ Query executed successfully. | MOLAP (14.0 RTM) | MOLAP\MOLAP (57) | Hotels | 00:00:01 | 14 rows

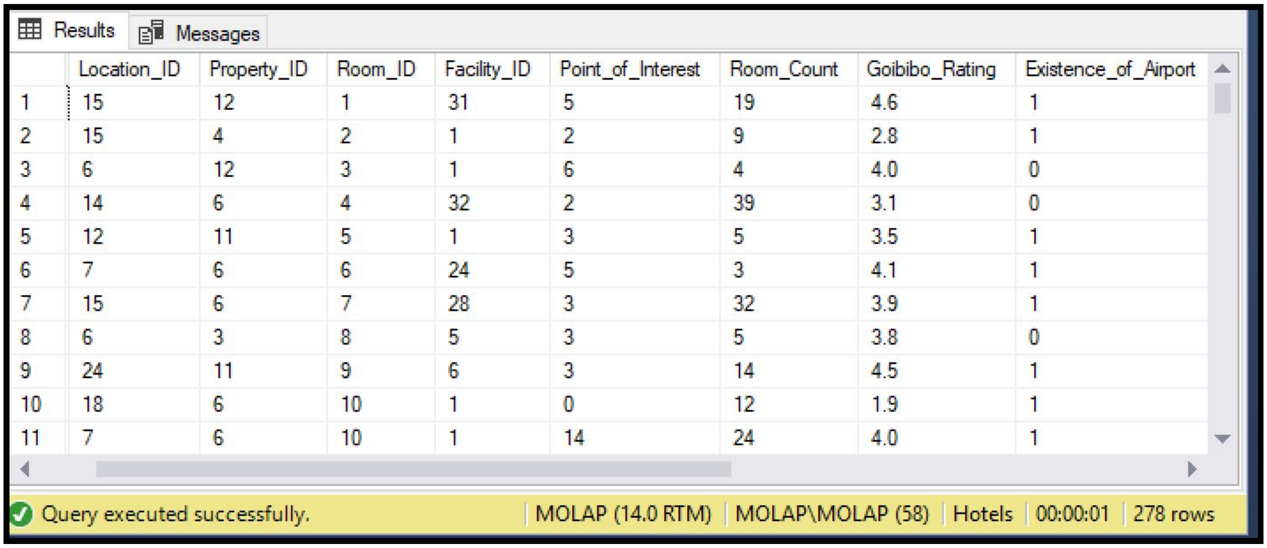
Figure 6:Property dimension table



	Room_ID	Room_Type
1	1	1 Bedroom
2	2	2 Bed Room
3	3	2 Room Apartment
4	4	A/C Deluxe Green
5	5	AC Cottage
6	6	AC Dormitory
7	7	AC Double Bedded Room
8	8	AC Double Room
9	9	AC Luxury Cottage
10	10	AC Room
11	11	AC Suite Room
12	12	Alpine Tent

Query executed successfully. | MOLAP (14.0 RTM) | MOLAP\MOLAP (59) | Hotels | 00:00:01 | 94 rows

Figure 7: Room dimension table

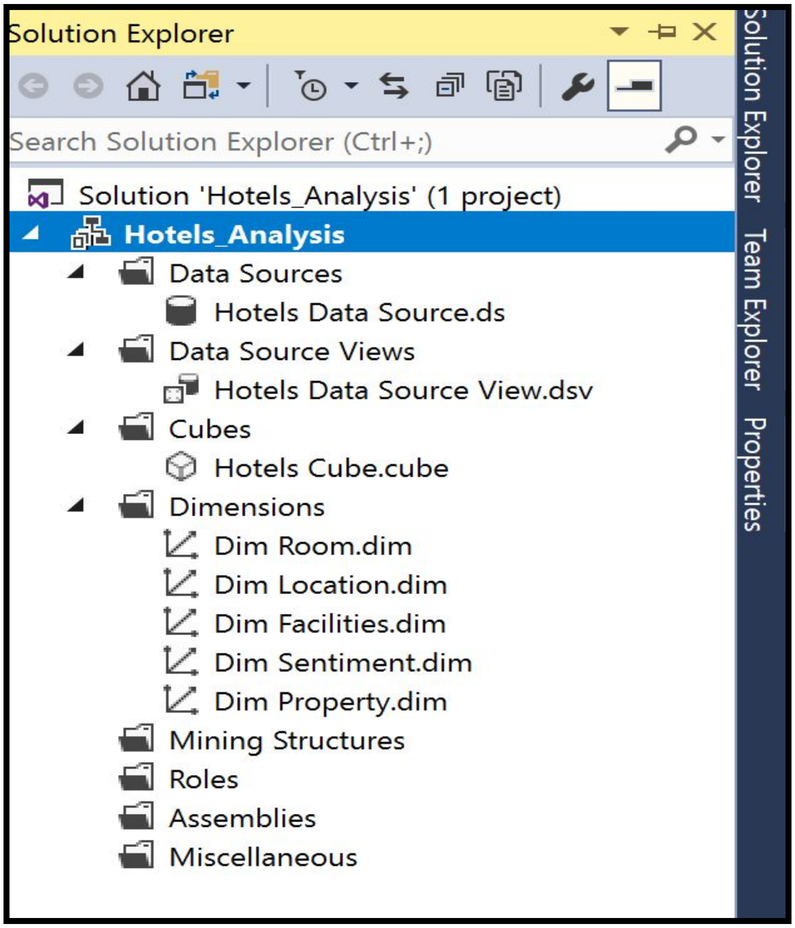


	Location_ID	Property_ID	Room_ID	Facility_ID	Point_of_Interest	Room_Count	Goibibo_Rating	Existence_of_Airport
1	15	12	1	31	5	19	4.6	1
2	15	4	2	1	2	9	2.8	1
3	6	12	3	1	6	4	4.0	0
4	14	6	4	32	2	39	3.1	0
5	12	11	5	1	3	5	3.5	1
6	7	6	6	24	5	3	4.1	1
7	15	6	7	28	3	32	3.9	1
8	6	3	8	5	3	5	3.8	0
9	24	11	9	6	3	14	4.5	1
10	18	6	10	1	0	12	1.9	1
11	7	6	10	1	14	24	4.0	1

Query executed successfully. | MOLAP (14.0 RTM) | MOLAP\MOLAP (58) | Hotels | 00:00:01 | 278 rows

Figure 8:Fact table

SQL server analysis services (SSAS): Finally, data warehouse is ready with populated dimension and fact tables. Next, cube is built for analysis purposes using SSAS. Data sources, data source views, dimensions and cube are created using SSAS. In this project, hotels datasources, hotels data source view, hotels cube and dimensions are created. The below screenshot provides the details of the cube.



- Solution 'Hotels_Analysis' (1 project)
  - Hotels_Analysis
    - Data Sources
      - Hotels Data Source.ds
    - Data Source Views
      - Hotels Data Source View.dsv
    - Cubes
      - Hotels Cube.cube
    - Dimensions
      - Dim Room.dim
      - Dim Location.dim
      - Dim Facilities.dim
      - Dim Sentiment.dim
      - Dim Property.dim
    - Mining Structures
    - Roles
    - Assemblies
    - Miscellaneous

Figure 9: SSAS window

As shown in Figure 10, we can clearly say data warehouse is of star schema model. Primary keys of dimension tables became the foreign keys in fact table. Data Source View of the Hotels Data warehouse is shown below.

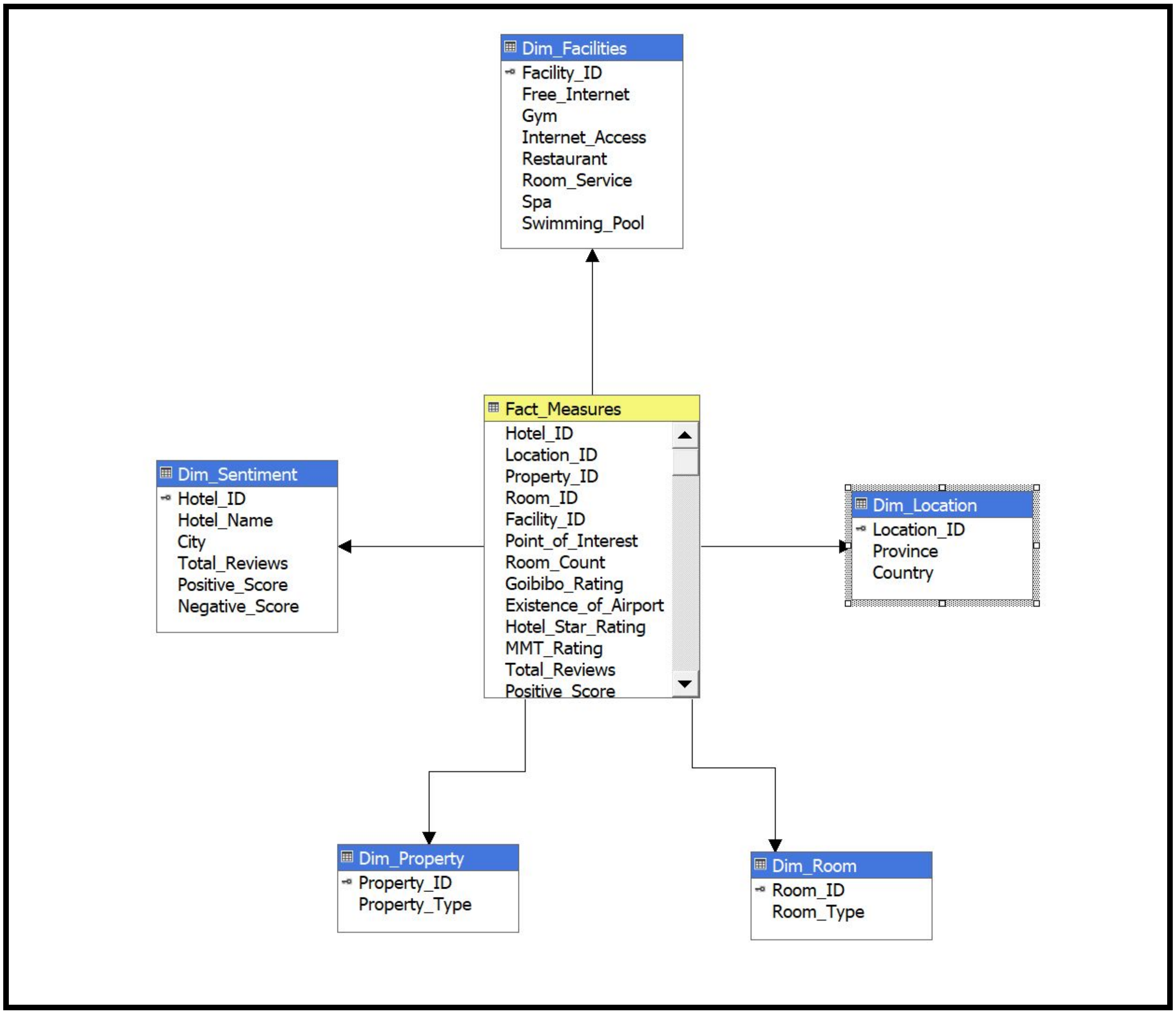
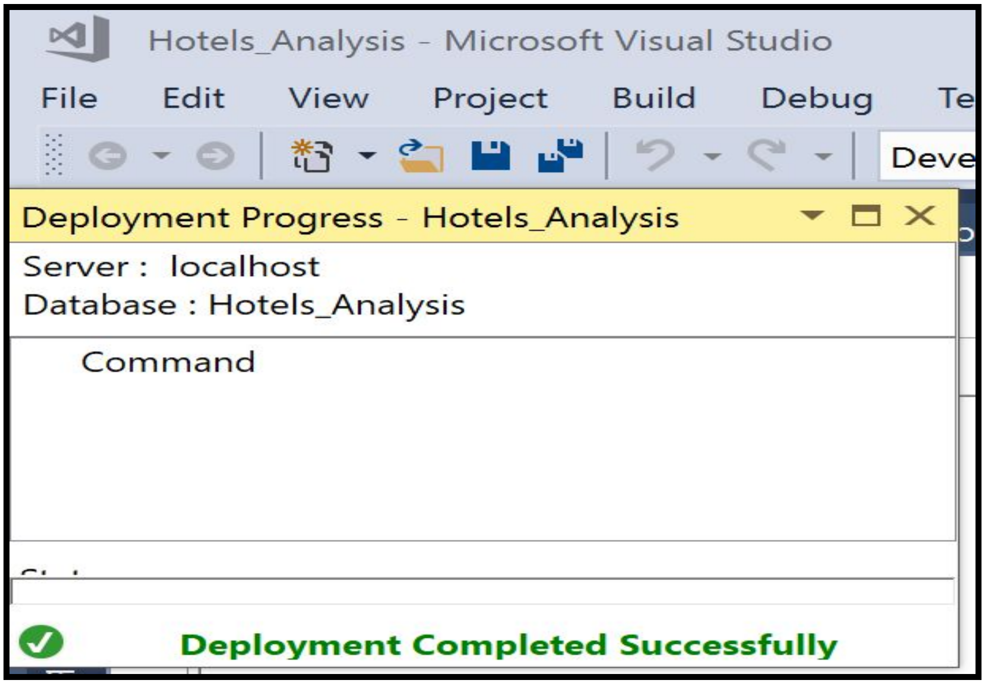


Figure 10: Star schema of project

Next step in the process is cube deployment. The screenshot of cube deployment is shown in Figure 10. We can observe that cube is deployed successfully.



Deployment Progress - Hotels_Analysis

Server : localhost
Database : Hotels_Analysis

Command

Deployment Completed Successfully

Figure 11:Cube deployment

Extraction,Transformation and Loading (ETL):

The data fetched from different data sources in not properly cleaned. In general, the real world data is full of redundancies and dirty. We need to clean the data properly in order to design a good data warehouse. So ETL is required to clean and process the data.

Automatic Extraction: In this project, the data is collected from three different sources. One of the structured data set is from **data.world** and another is from **Kaggle**. These two are in CSV format. Unstructured data set is constructed by scrapping reviews from TripAdvisor web links. The reviews were scrapped and sentiment score was calculated using R code.

Automated Cleaning: The extracted data sets from three data sources are **automatically cleaned using R programming**. Structured data sets which are in CSV format are cleaned, transformed and merged in one file. The R code for cleaning structured datasets is given in the appendix.

Automatic Transformation: Structured data sets are cleaned and transformed using R programming. **In these datasets, hotel facilities, point of interests, hotel overview**

are analyzed and transformed into proper format using R. As mentioned above, the reviews (for unstructured data) web scrapped from tripadvisor links and stored in CSV file by hotel wise. The reviews are of string data types and sentiment analysis is done in order to calculate sentiment score using R **[2]** (here I changed the code according to my needs). The script for cleaning, transformation of data sets is given in the appendix.

Automatic Loading and Cube Deployment: Finally, dimension tables and fact table are loaded into data warehouse automatically using SSIS. The cube is deployed in SSIS package itself.

Business Intelligence Queries:

Query 1: Finding two major reasons for "Low Standard" in hotel star rating.

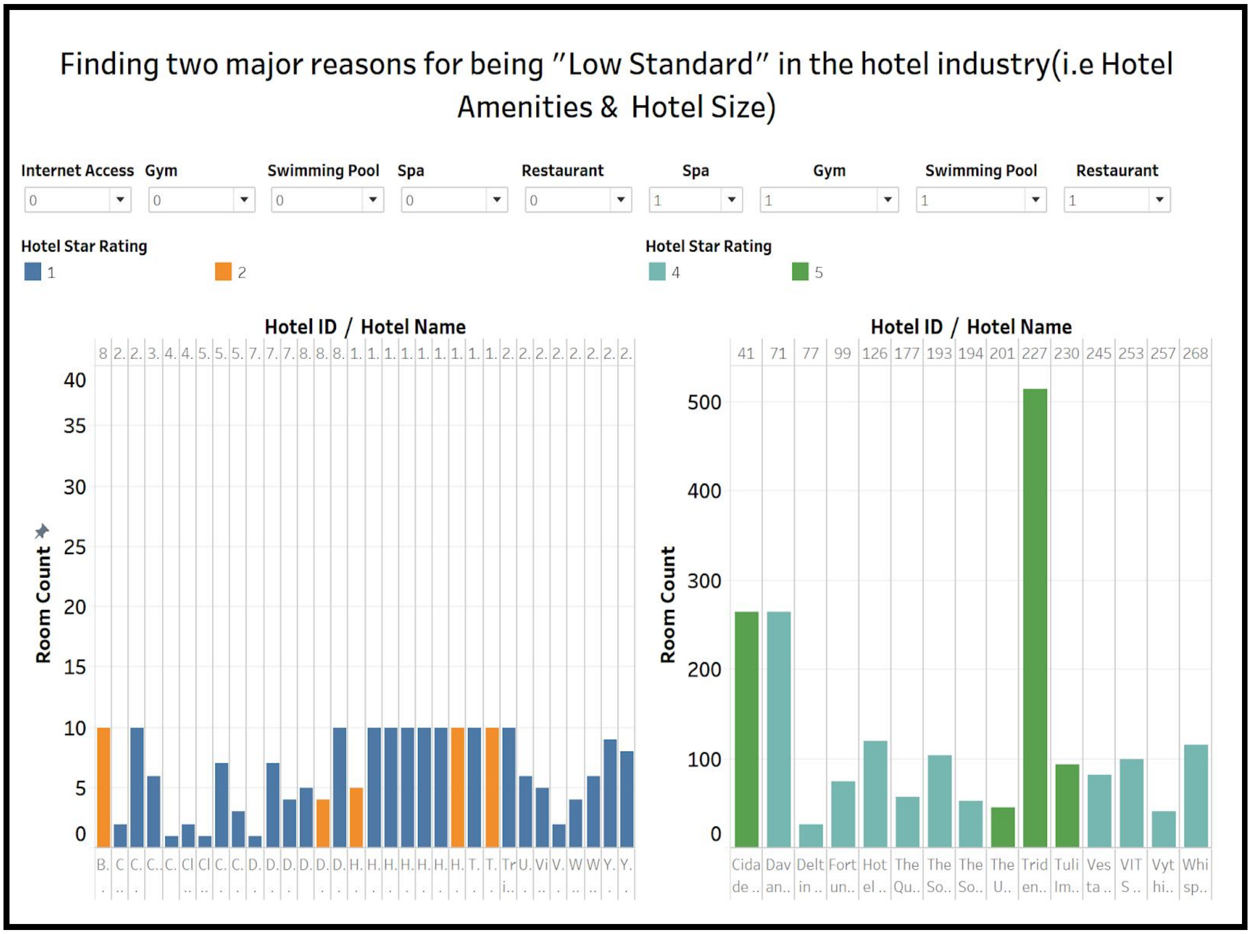


Figure 12: Query 1

In Hotels, we have different star ratings like 5 star hotel, 4 star hotel etc. From this data sets, the low standard of hotel rating is mainly due to two reasons.

In **Figure 12,** we can observe that 1 star and 2 star hotels have no facilities like internet access, gym, spa, swimming pool and restaurant. In the figure, ‘0’ in the filter indicates the facility is not provided. The room count (number of rooms in hotel < 10) is also quite less compared to 4 star and 5 star hotels. Coming to 4 star and 5 star hotels, they have all amenities like gym, spa, swimming pool and restaurant. ‘1’ in the filter indicates, the facility is provided. Comparatively, room count (number of rooms in hotel > 50) is more than 1 star and 2 star hotels. Therefore, **we can conclude that star rating of hotels depends upon the amenities they provide and how big the hotel is.**

Query 2: According to traveller reviews, which province has to work more on Tourism in order to contribute to economy?

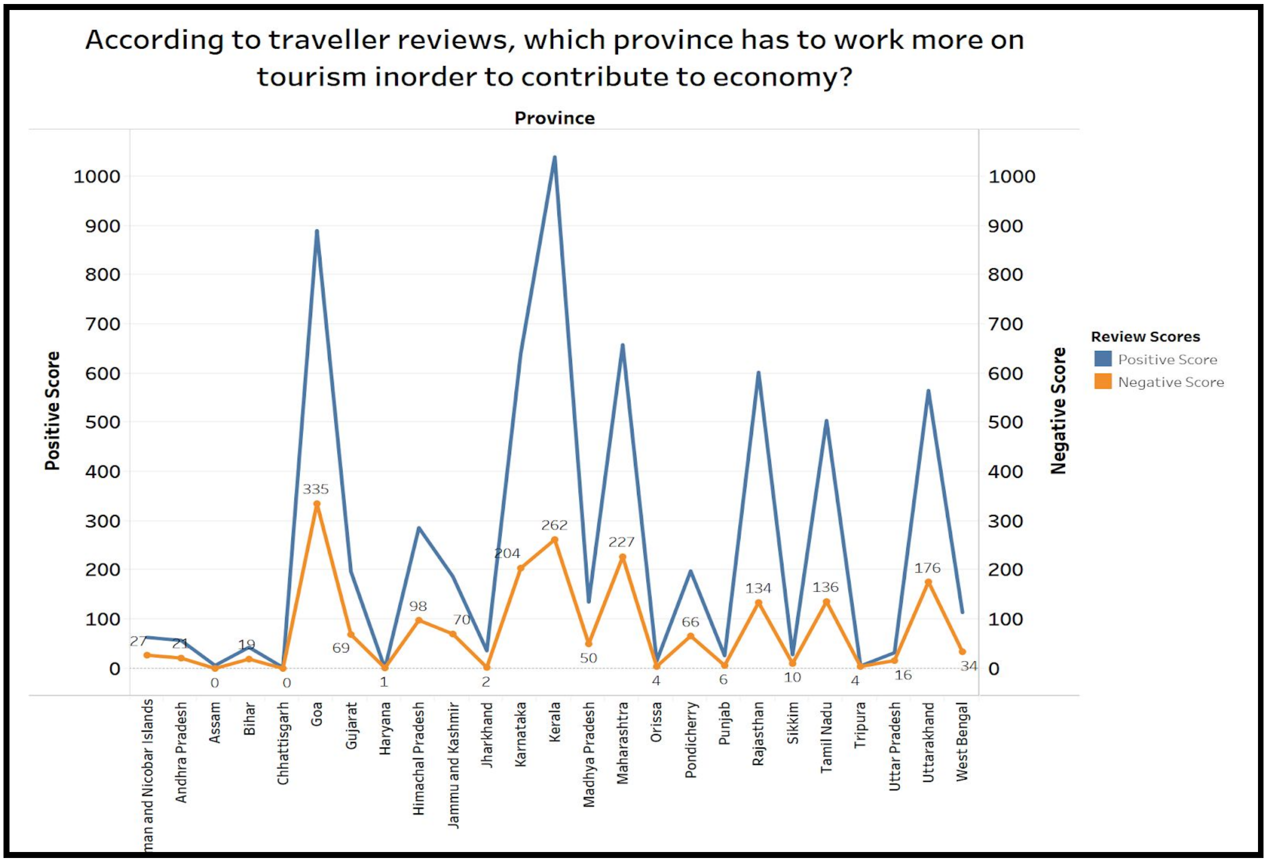


Figure 13: Query 2

In **Figure 13**, Goa has highest negative score (335) among all provinces. Positive score is indicated in orange color and negative score is indicated in blue color. These scores are obtained from tripadvisor hotel web links, where reviews are web scraped and analyzed. So, **Goa has to work more on traveller reviews which helps to increase tourism which in turn helps to improve country's economy.**

Query 3: Is there any relationship between user ratings and existence of airport close to hotel?

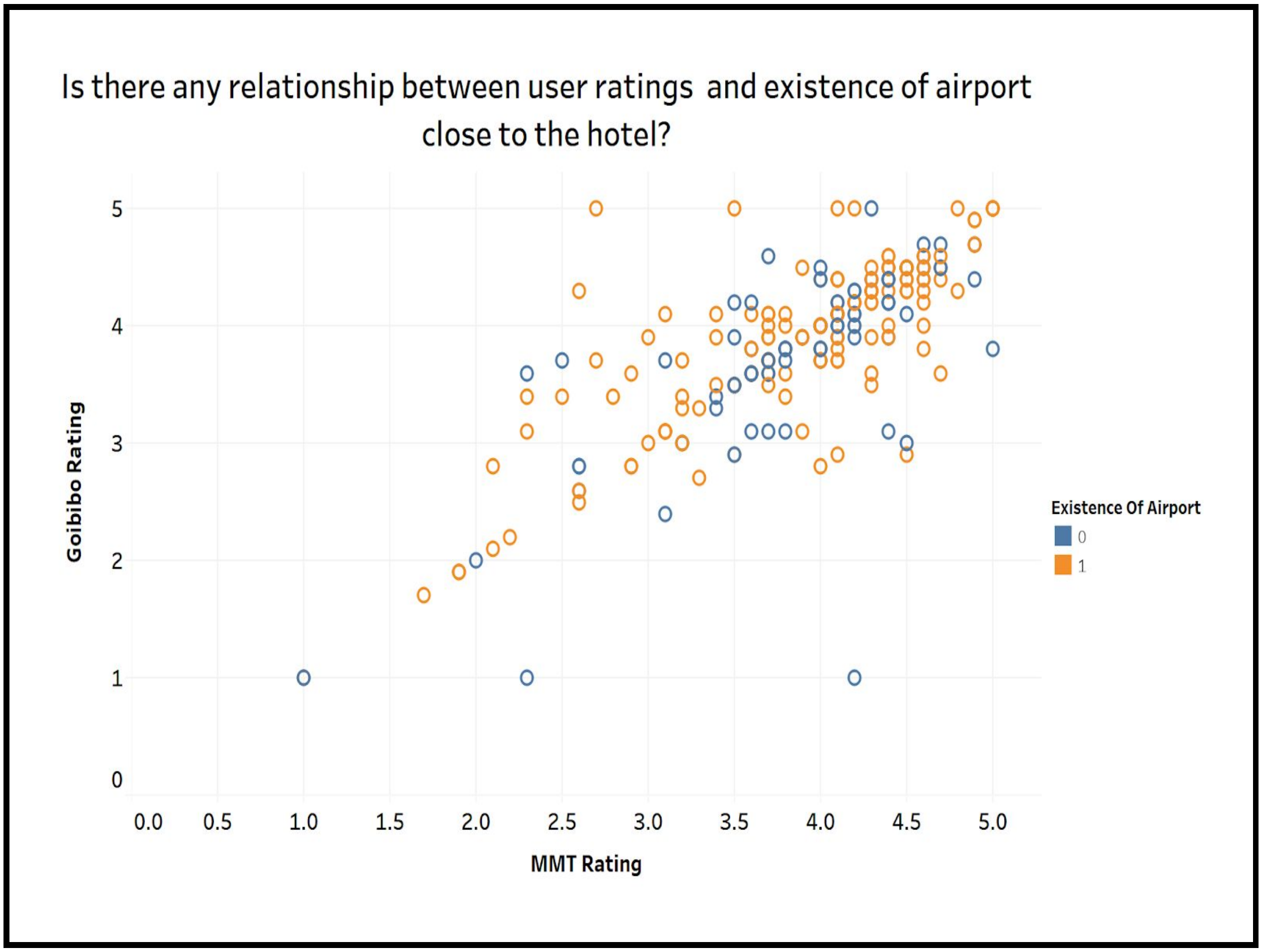


Figure 14: Query 3

These ratings are taken from two structured data sources (Makemytrip and Goibibo). Goibibo rating and makemytrip rating are the user ratings of the hotels on two different online travel sites. Here, **hotels having airport feasibility are good rated compared**

to the hotels without airport feasibility by the users. Orange color indicates the hotels with airport feasibility and blue color indicates the hotels without airport feasibility.

The other scenario we can observe here is, there is no big difference between these two ratings even though they are from two different travel websites. **From the scatter plot, we can observe that ratings are strongly correlated with each other.**

Query 4: Does the point of interests around the hotel impacted property type?

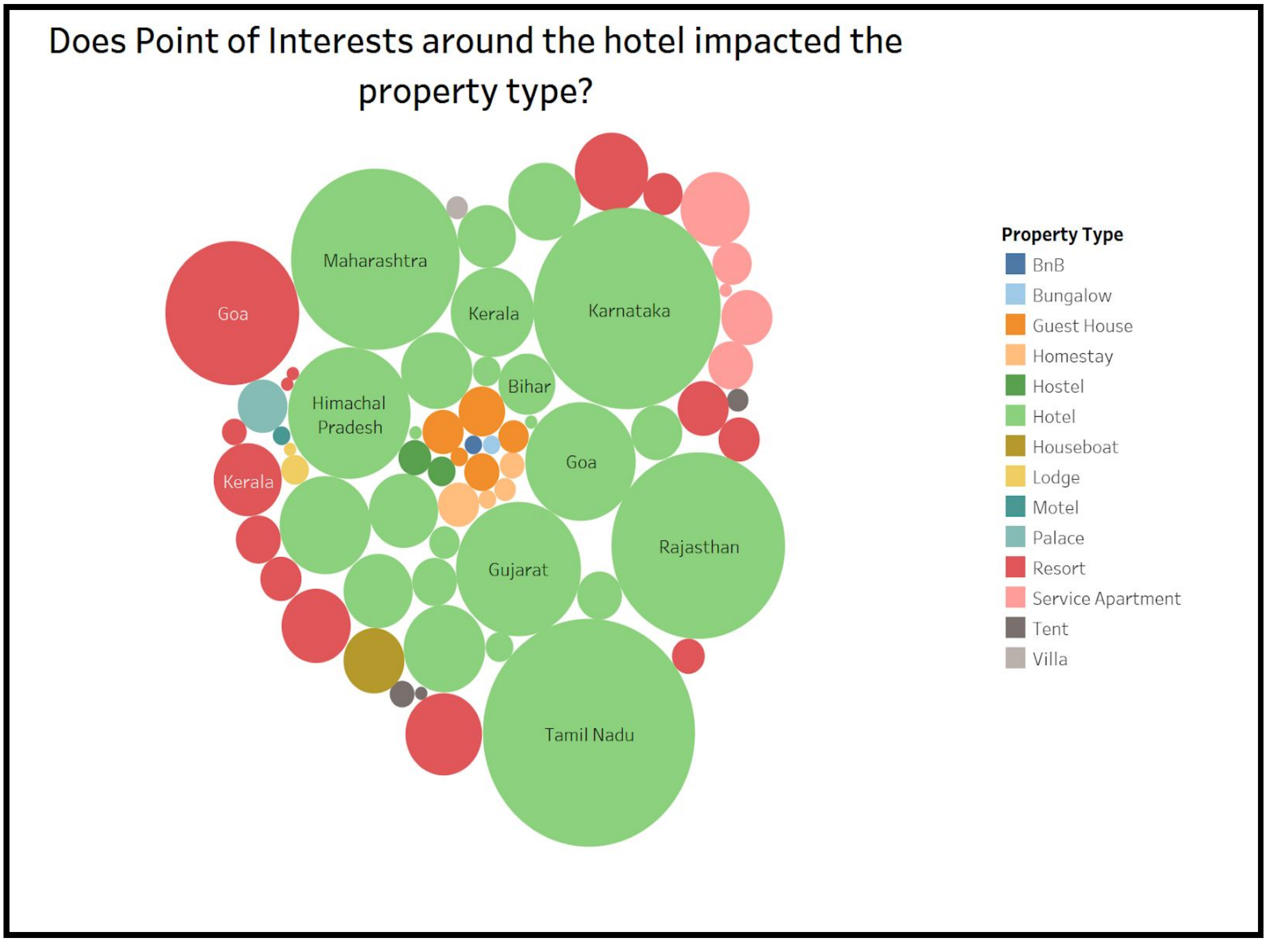


Figure 15: Query 4

The size of circle represents the number of point of interests around the hotel. From the above figure, we can observe that Tamilnadu have more tourist attractions compared to other states. Green color indicates the type of property is a hotel. We can observe that, where tourist attractions are more, the property type is a hotel. **So, we can conclude that hotels are preferred property type if tourist attractions are more.**

Conclusion

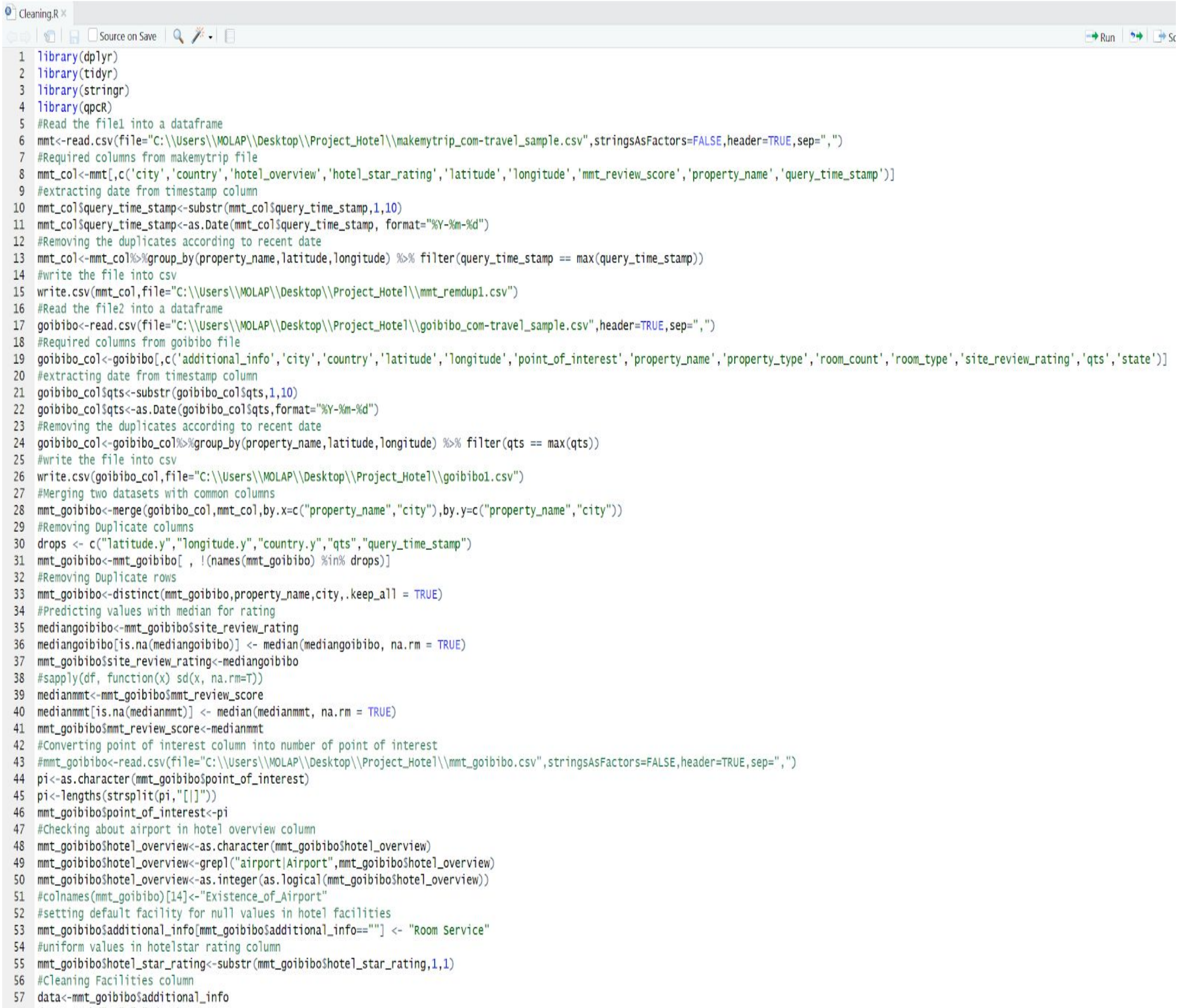
In this project, hotels data warehouse, business intelligence insights are provided which helps to enhance the profit, customer experience and working on loopholes. First query helps to work on the improvement of hotel standard. Second query helps to deal with the negative reviews of travellers which helps to enhance customer services. From third query, it is clear that user ratings does not depend upon the hotel services alone, but also other factors like airport feasibility. Fourth query concludes that the hotel is the most preferred property type based on tourist attractions.

References:

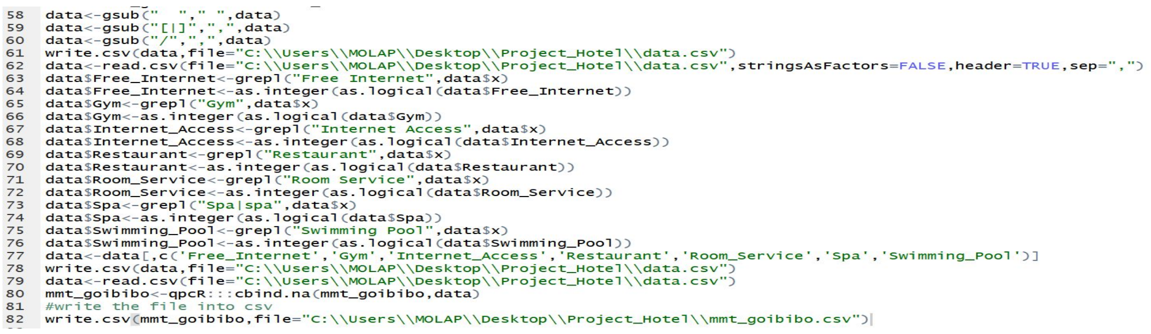
[1] Kimball, R. and Ross, M., 2011. The data warehouse toolkit: the complete guide to dimensional modeling. John Wiley & Sons.

[2] R script for sentiment Analysis ,
“https://github.com/agrawal-priyank/Web-Scraper-Sentiment-Analysis-TripAdvisor”

Appendix: Cleaning, transformation of structured datasets screenshot:

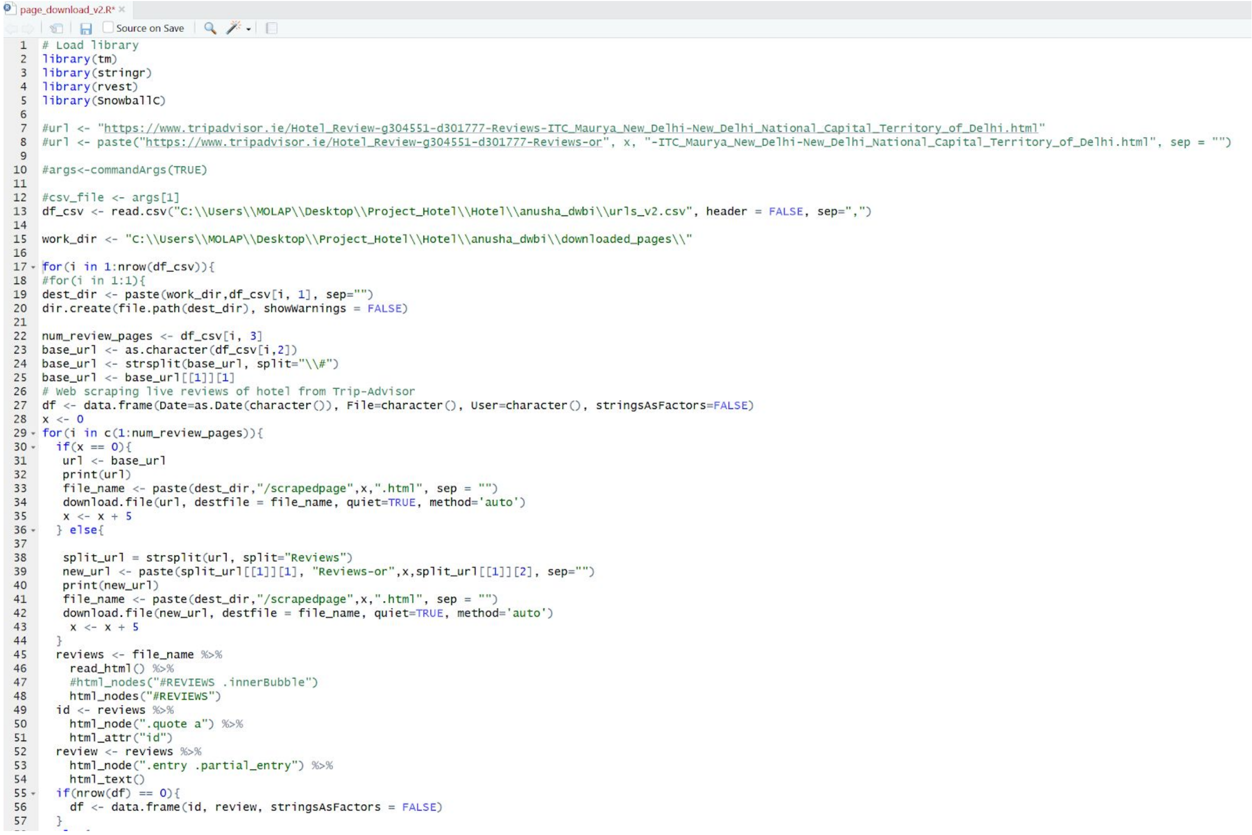


```r
library(dplyr)
library(tidyr)
library(stringr)
library(qpcR)
#Read the file1 into a dataframe
mmt<-read.csv(file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\makemytrip_com-travel_sample.csv",stringsAsFactors=FALSE,header=TRUE,sep=",")
#Required columns from makemytrip file
mmt_col<-mmt[,c('city','country','hotel_overview','hotel_star_rating','latitude','longitude','mmt_review_score','property_name','query_time_stamp')]
#extracting date from timestamp column
mmt_col$query_time_stamp<-substr(mmt_col$query_time_stamp,1,10)
mmt_col$query_time_stamp<-as.Date(mmt_col$query_time_stamp, format="%Y-%m-%d")
#Removing the duplicates according to recent date
mmt_col<-mmt_col%>%group_by(property_name,latitude,longitude) %>% filter(query_time_stamp == max(query_time_stamp))
#write the file into csv
write.csv(mmt_col,file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\mmt_remdup1.csv")
#Read the file2 into a dataframe
goibibo<-read.csv(file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\goibibo_com-travel_sample.csv",header=TRUE,sep=",")
#Required columns from goibibo file
goibibo_col<-goibibo[,c('additional_info','city','country','latitude','longitude','point_of_interest','property_name','property_type','room_count','room_type','site_review_rating','qts','state')]
#extracting date from timestamp column
goibibo_col$qts<-substr(goibibo_col$qts,1,10)
goibibo_col$qts<-as.Date(goibibo_col$qts,format="%Y-%m-%d")
#Removing the duplicates according to recent date
goibibo_col<-goibibo_col%>%group_by(property_name,latitude,longitude) %>% filter(qts == max(qts))
#write the file into csv
write.csv(goibibo_col,file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\goibibo1.csv")
#Merging two datasets with common columns
mmt_goibibo<-merge(goibibo_col,mmt_col,by.x=c("property_name","city"),by.y=c("property_name","city"))
#Removing Duplicate columns
drops <- c("latitude.y","longitude.y","country.y","qts","query_time_stamp")
mmt_goibibo<-mmt_goibibo[ , !(names(mmt_goibibo) %in% drops)]
#Removing Duplicate rows
mmt_goibibo<-distinct(mmt_goibibo,property_name,city,.keep_all = TRUE)
#Predicting values with median for rating
mediangoibibo<-mmt_goibibo$site_review_rating
mediangoibibo[is.na(mediangoibibo)] <- median(mediangoibibo, na.rm = TRUE)
mmt_goibibo$site_review_rating<-mediangoibibo
#sapply(df, function(x) sd(x, na.rm=T))
medianmmt<-mmt_goibibo$mmt_review_score
medianmmt[is.na(medianmmt)] <- median(medianmmt, na.rm = TRUE)
mmt_goibibo$mmt_review_score<-medianmmt
#Converting point of interest column into number of point of interest
#mmt_goibibo<-read.csv(file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\mmt_goibibo.csv",stringsAsFactors=FALSE,header=TRUE,sep=",")
pi<-as.character(mmt_goibibo$point_of_interest)
pi<-lengths(strsplit(pi,"[|]"))
mmt_goibibo$point_of_interest<-pi
#Checking about airport in hotel overview column
mmt_goibibo$hotel_overview<-as.character(mmt_goibibo$hotel_overview)
mmt_goibibo$hotel_overview<-grepl("airport|Airport",mmt_goibibo$hotel_overview)
mmt_goibibo$hotel_overview<-as.integer(as.logical(mmt_goibibo$hotel_overview))
#colnames(mmt_goibibo)[14]<-"Existence_of_Airport"
#setting default facility for null values in hotel facilities
mmt_goibibo$additional_info[mmt_goibibo$additional_info==""] <- "Room Service"
#uniform values in hotelstar rating column
mmt_goibibo$hotel_star_rating<-substr(mmt_goibibo$hotel_star_rating,1,1)
#Cleaning Facilities column
data<-mmt_goibibo$additional_info
```

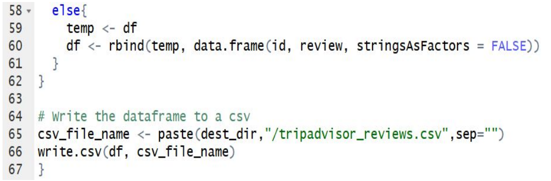


```r
data<-gsub("  "," ",data)
data<-gsub("[|]",",",data)
data<-gsub("/",",",data)
write.csv(data,file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\data.csv")
data<-read.csv(file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\data.csv",stringsAsFactors=FALSE,header=TRUE,sep=",")
data$Free_Internet<-grepl("Free Internet",data$x)
data$Free_Internet<-as.integer(as.logical(data$Free_Internet))
data$Gym<-grepl("Gym",data$x)
data$Gym<-as.integer(as.logical(data$Gym))
data$Internet_Access<-grepl("Internet Access",data$x)
data$Internet_Access<-as.integer(as.logical(data$Internet_Access))
data$Restaurant<-grepl("Restaurant",data$x)
data$Restaurant<-as.integer(as.logical(data$Restaurant))
data$Room_Service<-grepl("Room Service",data$x)
data$Room_Service<-as.integer(as.logical(data$Room_Service))
data$Spa<-grepl("Spa|spa",data$x)
data$Spa<-as.integer(as.logical(data$Spa))
data$Swimming_Pool<-grepl("Swimming Pool",data$x)
data$Swimming_Pool<-as.integer(as.logical(data$Swimming_Pool))
data<-data[,c('Free_Internet','Gym','Internet_Access','Restaurant','Room_Service','Spa','Swimming_Pool')]
write.csv(data,file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\data.csv")
data<-read.csv(file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\data.csv")
mmt_goibibo<-qpcR:::cbind.na(mmt_goibibo,data)
#write the file into csv
write.csv(mmt_goibibo,file="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\mmt_goibibo.csv")
```

Web scraping reviews in tripadvisor:

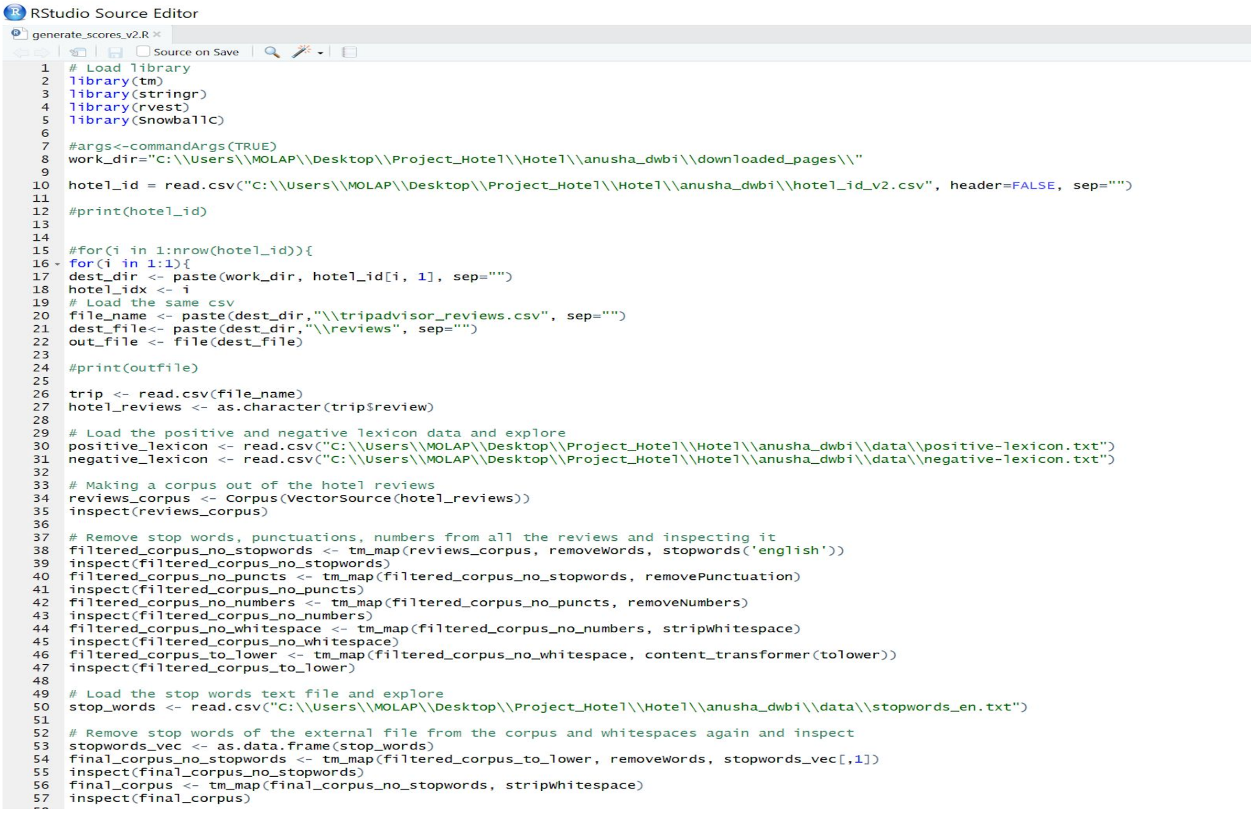


```r
# Load library
library(tm)
library(stringr)
library(rvest)
library(SnowballC)

#url <- "https://www.tripadvisor.ie/Hotel_Review-g304551-d301777-Reviews-ITC_Maurya_New_Delhi-New_Delhi_National_Capital_Territory_of_Delhi.html"
#url <- paste("https://www.tripadvisor.ie/Hotel_Review-g304551-d301777-Reviews-or", x, "-ITC_Maurya_New_Delhi-New_Delhi_National_Capital_Territory_of_Delhi.html", sep = "")

#args<-commandArgs(TRUE)

#csv_file <- args[1]
df_csv <- read.csv("C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\urls_v2.csv", header = FALSE, sep=",")

work_dir <- "C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\downloaded_pages\\"

for(i in 1:nrow(df_csv)){
#for(i in 1:1){
dest_dir <- paste(work_dir,df_csv[i, 1], sep="")
dir.create(file.path(dest_dir), showWarnings = FALSE)

num_review_pages <- df_csv[i, 3]
base_url <- as.character(df_csv[i,2])
base_url <- strsplit(base_url, split="\\#")
base_url <- base_url[[1]][1]
# Web scraping live reviews of hotel from Trip-Advisor
df <- data.frame(Date=as.Date(character()), File=character(), User=character(), stringsAsFactors=FALSE)
x <- 0
for(i in c(1:num_review_pages)){
  if(x == 0){
   url <- base_url
   print(url)
   file_name <- paste(dest_dir,"/scrapedpage",x,".html", sep = "")
   download.file(url, destfile = file_name, quiet=TRUE, method='auto')
   x <- x + 5
  } else{

   split_url = strsplit(url, split="Reviews")
   new_url <- paste(split_url[[1]][1], "Reviews-or",x,split_url[[1]][2], sep="")
   print(new_url)
   file_name <- paste(dest_dir,"/scrapedpage",x,".html", sep = "")
   download.file(new_url, destfile = file_name, quiet=TRUE, method='auto')
    x <- x + 5
  }
  reviews <- file_name %>%
    read_html() %>%
    #html_nodes("#REVIEWS .innerBubble")
    html_nodes("#REVIEWS")
  id <- reviews %>%
    html_node(".quote a") %>%
    html_attr("id")
  review <- reviews %>%
    html_node(".entry .partial_entry") %>%
    html_text()
  if(nrow(df) == 0){
    df <- data.frame(id, review, stringsAsFactors = FALSE)
  }
```

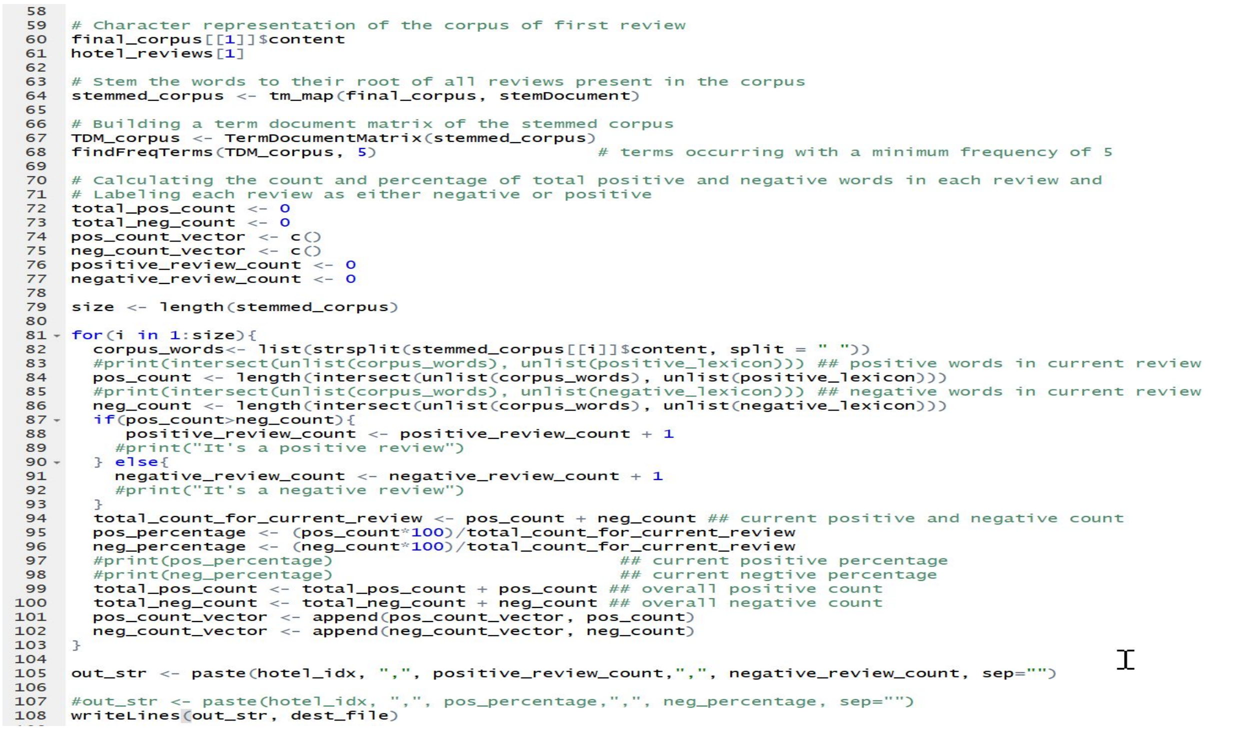


```r
  else{
    temp <- df
    df <- rbind(temp, data.frame(id, review, stringsAsFactors = FALSE))
  }
}

# Write the dataframe to a csv
csv_file_name <- paste(dest_dir,"/tripadvisor_reviews.csv",sep="")
write.csv(df, csv_file_name)
}
```

Sentiment Analysis:



```r
# Load library
library(tm)
library(stringr)
library(rvest)
library(SnowballC)

#args<-commandArgs(TRUE)
work_dir="C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\downloaded_pages\\"

hotel_id = read.csv("C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\hotel_id_v2.csv", header=FALSE, sep="")

#print(hotel_id)


#for(i in 1:nrow(hotel_id)){
for(i in 1:1){
dest_dir <- paste(work_dir, hotel_id[i, 1], sep="")
hotel_idx <- i
# Load the same csv
file_name <- paste(dest_dir,"\\tripadvisor_reviews.csv", sep="")
dest_file<- paste(dest_dir,"\\reviews", sep="")
out_file <- file(dest_file)

#print(outfile)

trip <- read.csv(file_name)
hotel_reviews <- as.character(trip$review)

# Load the positive and negative lexicon data and explore
positive_lexicon <- read.csv("C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\data\\positive-lexicon.txt")
negative_lexicon <- read.csv("C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\data\\negative-lexicon.txt")

# Making a corpus out of the hotel reviews
reviews_corpus <- Corpus(VectorSource(hotel_reviews))
inspect(reviews_corpus)

# Remove stop words, punctuations, numbers from all the reviews and inspecting it
filtered_corpus_no_stopwords <- tm_map(reviews_corpus, removeWords, stopwords('english'))
inspect(filtered_corpus_no_stopwords)
filtered_corpus_no_puncts <- tm_map(filtered_corpus_no_stopwords, removePunctuation)
inspect(filtered_corpus_no_puncts)
filtered_corpus_no_numbers <- tm_map(filtered_corpus_no_puncts, removeNumbers)
inspect(filtered_corpus_no_numbers)
filtered_corpus_no_whitespace <- tm_map(filtered_corpus_no_numbers, stripWhitespace)
inspect(filtered_corpus_no_whitespace)
filtered_corpus_to_lower <- tm_map(filtered_corpus_no_whitespace, content_transformer(tolower))
inspect(filtered_corpus_to_lower)

# Load the stop words text file and explore
stop_words <- read.csv("C:\\Users\\MOLAP\\Desktop\\Project_Hotel\\Hotel\\anusha_dwbi\\data\\stopwords_en.txt")

# Remove stop words of the external file from the corpus and whitespaces again and inspect
stopwords_vec <- as.data.frame(stop_words)
final_corpus_no_stopwords <- tm_map(filtered_corpus_to_lower, removeWords, stopwords_vec[,1])
inspect(final_corpus_no_stopwords)
final_corpus <- tm_map(final_corpus_no_stopwords, stripWhitespace)
inspect(final_corpus)
```



```r

# Character representation of the corpus of first review
final_corpus[[1]]$content
hotel_reviews[1]

# Stem the words to their root of all reviews present in the corpus
stemmed_corpus <- tm_map(final_corpus, stemDocument)

# Building a term document matrix of the stemmed corpus
TDM_corpus <- TermDocumentMatrix(stemmed_corpus)
findFreqTerms(TDM_corpus, 5)                    # terms occurring with a minimum frequency of 5

# Calculating the count and percentage of total positive and negative words in each review and
# Labeling each review as either negative or positive
total_pos_count <- 0
total_neg_count <- 0
pos_count_vector <- c()
neg_count_vector <- c()
positive_review_count <- 0
negative_review_count <- 0

size <- length(stemmed_corpus)

for(i in 1:size){
  corpus_words<- list(strsplit(stemmed_corpus[[i]]$content, split = " "))
  #print(intersect(unlist(corpus_words), unlist(positive_lexicon))) ## positive words in current review
  pos_count <- length(intersect(unlist(corpus_words), unlist(positive_lexicon)))
  #print(intersect(unlist(corpus_words), unlist(negative_lexicon))) ## negative words in current review
  neg_count <- length(intersect(unlist(corpus_words), unlist(negative_lexicon)))
  if(pos_count>neg_count){
     positive_review_count <- positive_review_count + 1
    #print("It's a positive review")
  } else{
    negative_review_count <- negative_review_count + 1
    #print("It's a negative review")
  }
  total_count_for_current_review <- pos_count + neg_count ## current positive and negative count
  pos_percentage <- (pos_count*100)/total_count_for_current_review
  neg_percentage <- (neg_count*100)/total_count_for_current_review
  #print(pos_percentage)                          ## current positive percentage
  #print(neg_percentage)                          ## current negtive percentage
  total_pos_count <- total_pos_count + pos_count ## overall positive count
  total_neg_count <- total_neg_count + neg_count ## overall negative count
  pos_count_vector <- append(pos_count_vector, pos_count)
  neg_count_vector <- append(neg_count_vector, neg_count)
}

out_str <- paste(hotel_idx, ",", positive_review_count,",", negative_review_count, sep="")

#out_str <- paste(hotel_idx, ",", pos_percentage,",", neg_percentage, sep="")
writeLines(out_str, dest_file)
```