World Tourism Data Warehouse

Data Warehousing and Business Intelligence Project

Siddharth Sharma
M.Sc. in Data Analytics
National College of Ireland
X16148371@student.ncirl.ie

INTRODUCTION

Business scenarios affecting today's marketplace are growing exponentially and need of Data Analysis and Data warehousing is increasing day by day for every organization. In the past few years there has been an upsurge in generation of data someone needs to analyze it and extract information which further can be transformed into knowledge. Data Warehousing helps to extract tons of data, transform it into meaningful statistics data and finally use it for influencing the decision making of an organization.
In this project I explore the realms of world tourism, what is wanderlust? Why do people travel so much? Where do they go? The answer lies in my 'Tourism Data Warehouse'. I have made my analysis based on five years of data (2011-2015) from various sources.
Approach for the project is as follows:

- Discovering of data sources.
- Establish a Data Warehouse
- ETL staging
- Deployment of ETL

DISCOVERING DATA SOURCES:
The following data sources have been used for analysis:

1. **Tourist Arrivals (Structured Dataset)**
   This dataset contains information of the tourists arriving in specific countries from 2011 to 2015.
   Source of data source: http://world-statistics.org/index-res.php?code=ST.INT.ARVL?name=International%20tourism,%20number%20of%20arrivals#top-result

2. **Tourists Departures (Structured Dataset)**
   This dataset contains information about the tourists departing from particular countries from 2011 to 2015. I have created a dumpy dataset from the website mockaroo for generating data for tourist departures.
   Source of data source: http://www.mockaroo.com

3. **Tourist Accommodation (Structured Dataset)**
   This dataset contains information about the number of accommodations for tourists in distinct countries from 2011 to 2015. The dataset contains the total number of accommodations for inbound and outbound tourists.
   Source of data source: - https://knoema.com/WTODB2017/world-tourism-organization-database-2017

4. **Twitter (Un-Structured Dataset)**
   I have fetched tweets about various countries which will fetch me data about how much people like a particular place using sentiment analysis. I have added the country code of each country so that I can concatenate with other datasets. Sentiment Analysis on Twitter helped me gain sight on which are the favorite countries people around the world visit fondly. Analysis has been done using several R packages in R-Studio. I have added country code using R for the distinct countries, for better connection between the datasets [1].

TECHNOLOGIES USED:
SSMS: - SQL Server Management Studio
SSIS: - SQL Server Integration Services
SSAS: - SQL Server Analytic Services
R: - Open Source programming language and software environment for statistical computing & graphics.
Tableau: - Interactive data visualization tool.

DATA WAREHOUSE ARCHITECTURE

I have used Kimball's bottom-up approach towards building the data warehouse. Inmon's top-down approach is not used because data marts are created after the data warehouse has been deployed which takes a large amount of time, and is not suitable for my project. Kimball's Method is better because of the following benefits:

- Operational Source Systems help in dealing with the performance and availability of the data [2].
- Extract, Transformation, and Load (ETL) system in Kimball's model is done through dimensional models which include the combing of data from multiple sources, cleaning the raw data gathered and normalizing the data [2]. It processes the queries faster and the required joins are simple for fetching data.
- Presentation Area is where data is demonstrated & stashed in form of a star schema or a snowflake schema which helps in building an agile, decentralized data warehouse [2].
- Business Intelligence Application helps the users to access the data via modeling tools [2].

The four- steps used in dimensional modelling [2]

1) Business process is selected
2) The Grain is declared
3) Dimensions are identified
4) The facts are pinned down

Grain used in this project is the number of tourists arriving in a specific country or departing to another country and these measures have been pinned down in the fact table. Four dimension tables have been used to populate the fact table which correspond to a measurement event.

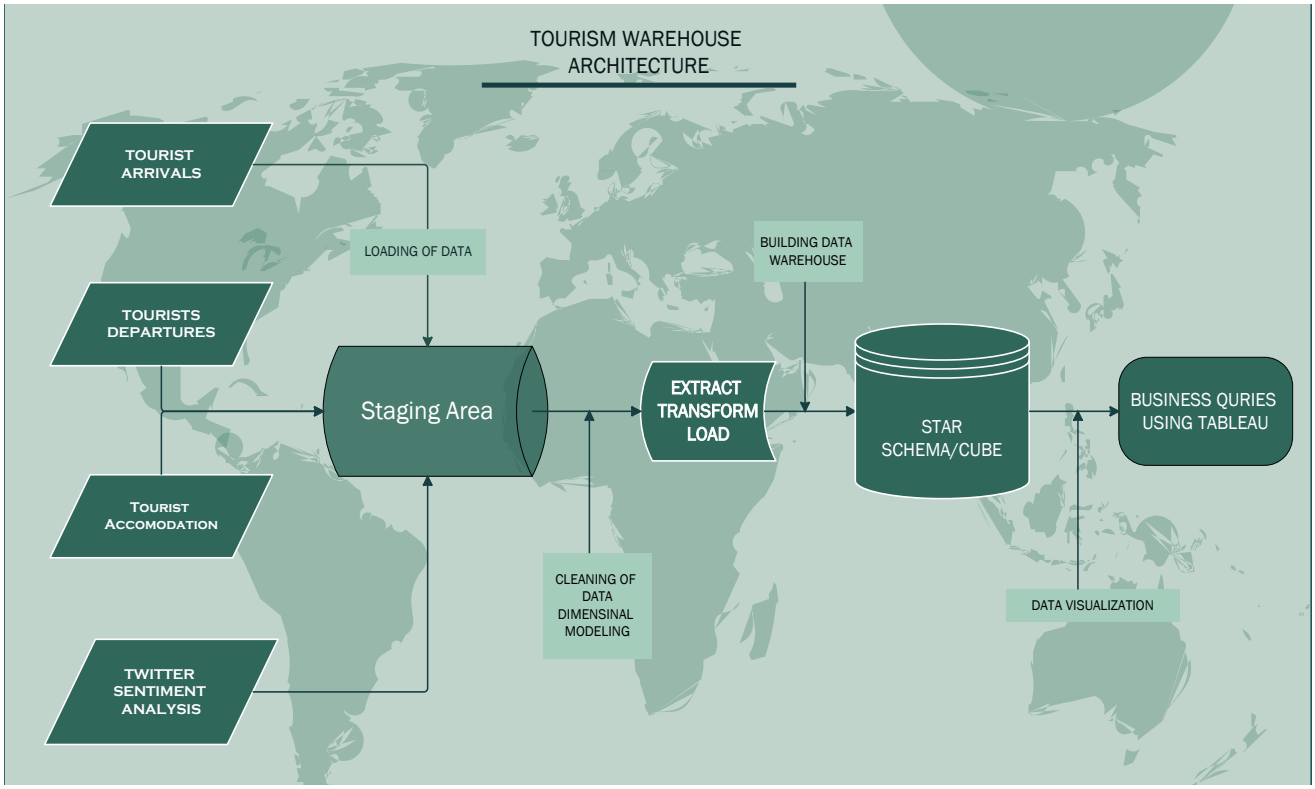

Figure 1

STAGING AREA:

Data has been loaded from different raw data sources into the appropriate tables. Data has been cleaned and the required columns that will be used in extraction, transformation & loading have been sorted out. All the sources have measurable values that will feed in the values to the data warehouse at a later stage.

DATA WAREHOUSE DATA MODEL

STAR SCHEMA:

This architecture is a fundamental data warehouse schema. And it is called so because it resembles to a star like structure. The fact table is at the nucleus of the schema while the dimensions are connected around the fact table, which typically contains all the measurable values from the raw data files and the foreign keys from the dimension tables.

DIMENSION TABLES:

The designated primary keys are foreign keys to the fact table. They contain textual content associated with the business process. They describe the "who, what, where, when, how and why" corresponding to the business processes [2]. Dimension attributes serve as the primary source of query constrains and report labels [2]. The analytical power of any Data Warehouse/Business Intelligence environment is relative to the quality and insight of the attributes [2].

Benefits of using a Star Schema are as follows:

1) A star schema is a decent physical foundation for building an OLAP (Online Analytical Processing) cube since it is steadier to support reinforcements and recovery [2].
2) It has a faster Extraction, transformation & Loading(ETL) capabilities.
3) It is composed of less number of foreign keys and lower query complexity.

Star Schema for "Tourism Data Warehouse"

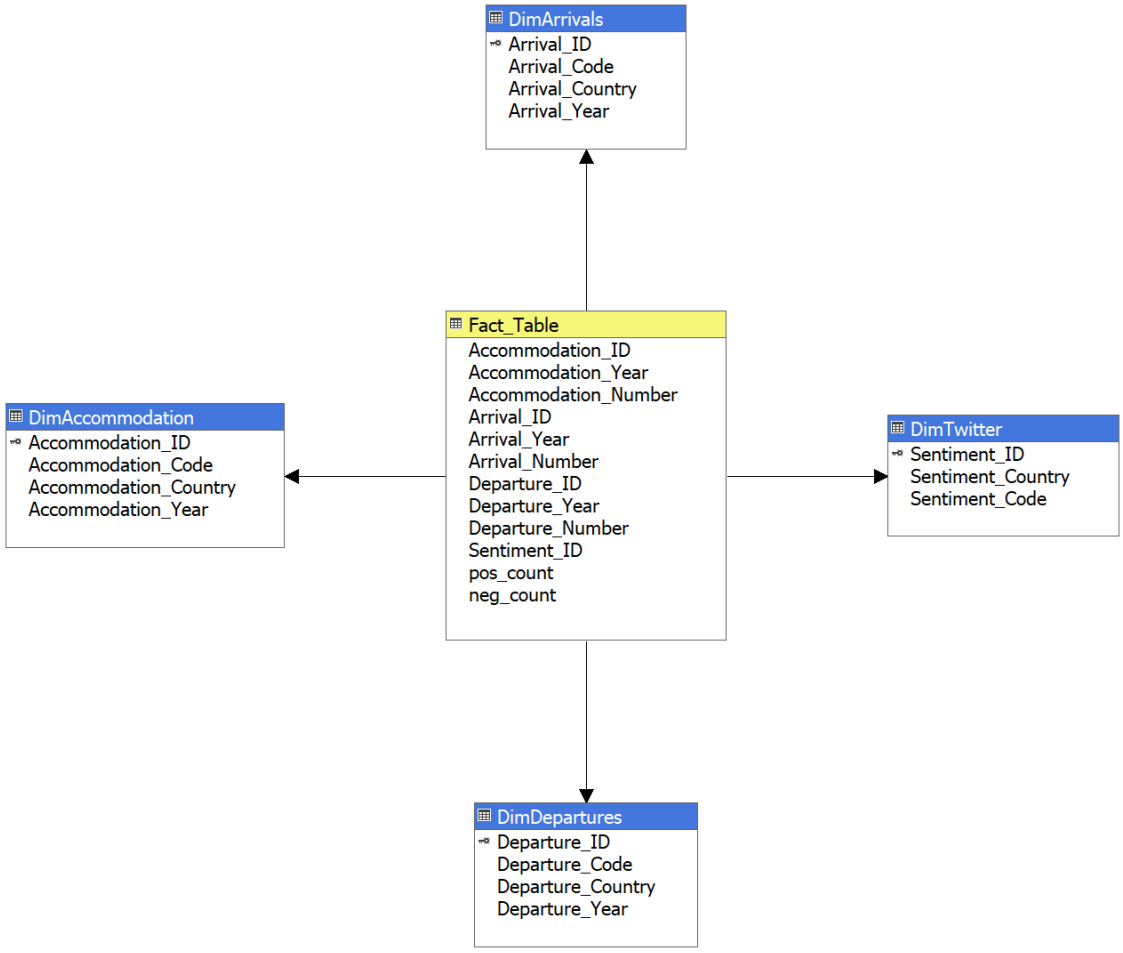

Figure 2

DESIGING OF DATA WAREHOUSE

Here are the four dimension tables used in the star schema:

1) **DimArrivals**
   This dimension contains the data that has been retrieved from world statistics database. It consists of arrival id's (Primary key), country codes, country names and the year in which the tourists travelled around the world. All the measurable values such as the number of arrivals per country have been included in the fact table.
2) **DimDepartures**
   This dimension contains the data that has been pulled out of world tourism Database. Here reside departure id's (Primary Key), departure codes, country names and the year which people preferred to travel around that sit idle at home.
   The number of departures
3) **DimAccommodation**
   This dimension contains the accommodation id's (Primary key), country names, country codes and the year specifying the number of accommodations in distinct countries which are available for all the visitors irrespective of outbound and inbound tourism.
4) **DimTwitter**
   This dimension is composed of the sentiment id's, country code, country names that have been fetched from the tweets using R for sentiment analysis which will be useful for the business quires.

In the dimension table's there exists a unique id for all the four dimension tables i.e. DimArrivals, DimDeparture, DimAccommodation & DimTwitter, they all are assigned a primary key which is used to join them to the fact table at a later stage of ETL process. The dimension tables used above are all interlinked and maneuvered to get the business queries.

- Dim Arrivals and Dim Departure are used to analyze the situation of tourism around the globe, and compare distinct countries on basis of time-series (2011-2015).
- DimAccommodation and Dim Arrivals have been used to probe on the housing facilities provided by various countries for the inbound and outbound tourists. In addition that the world's top destinations can be deduced.
- DimTwitter, DimArrivals and DimAccommodation are utilized to investigate the reviews of people around the world on their favorite destinations.

EXTRACT, TRANSFORM & LOAD

In order to construct a data warehouse, we have to extract the data from the sources, transform it according to the business processes defined at the start and then load them into the data warehouse. SSIS (SQL Server Integration Services) and SSMS (SQL Server Integration Services) have been put to use for extraction, transformation and loading of the data.

DATA EXTRACTION

Data has been extracted from all the datasets present in the database (SSMS). There are four datasets that are used: tourist's Arrival data, tourist's departure data, tourist's accommodation data and finally the Sentiment Analysis data from twitter. All the data has been extracted in .CSV format (From the structured data sources) so that data can be visualized appropriately and help us in lodging the data in the various dimensions.

The data from tweets, assembled via twitter has been cleaned & reconstructed into a genuine dataset using R (Code has been provided at the end of the document), as it is unstructured data.

I have added the country code using R for distinct countries so that connection can be made with other datasets in an appropriate way.

DATA TRANSFROMATION

Here data is cleaned and converted into purposeful information. I have used a number of lookups and changed the data type of a few variables to avoid the data conversion error. I have assembled all the data from the four data sources and used it to populate the dimension table and finally the fact table using lookups and made use of SQL Queries (Code has been mentioned at the end of the document). As there are limited number of data sources therefore cleaning of data cannot be augmented further.

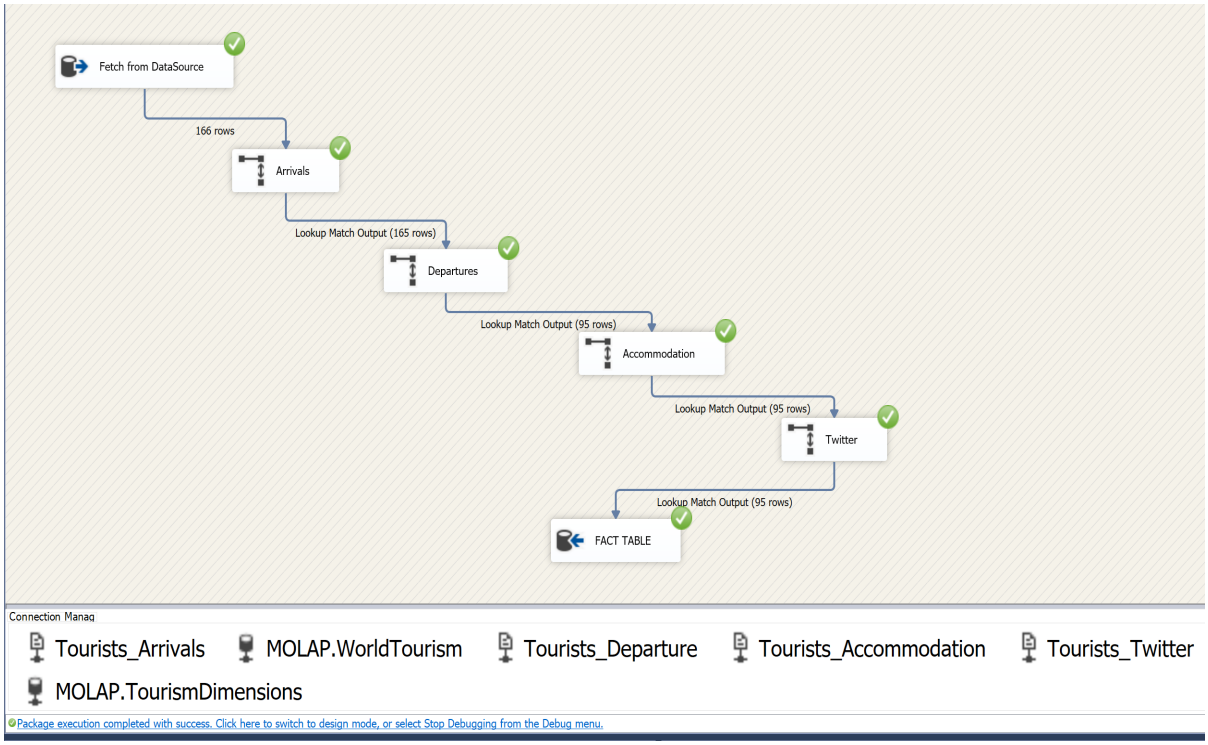

Figure 3

DATA LOADING

I have created two databases for the easy of processing. First one is WorldTourism for storing all the raw data then Tourism Dimensions for stocking data of the fact_table and the dimension tables. Firstly we load the data from external sources into the WorldTourism database which results in four raw data tables. Then data is loaded into TourismDimension database from the raw data tables which results in the creation of dimension tables. The data inserted into the dimension tables is limited to a few columns as all the measurable values will be present in the fact table and only descriptive / textual data is present in the dimension tables e.g. country names, country names etc.
Loading data into fact table is the last step in the Extraction, Transformation & Loading process (ETL). Fact table is populated via the dimension tables. The fact table is linked to the dimension tables via primary keys to particular data sets exist in the dimension tables, these primary keys act as foreign keys in the fact table, thus giving us the star schema.
After getting the star schema, cube is deployed using SSAS (SQL Server Analytic Services) through which our next section i.e. Applications of Data Warehouse (Business Intelligence Queries) is visualized through the help of an interactive data visualization tool Tableau.

Applications of Data Warehouse

1st Business Intelligence Query

Which is the most attractive place in the world?

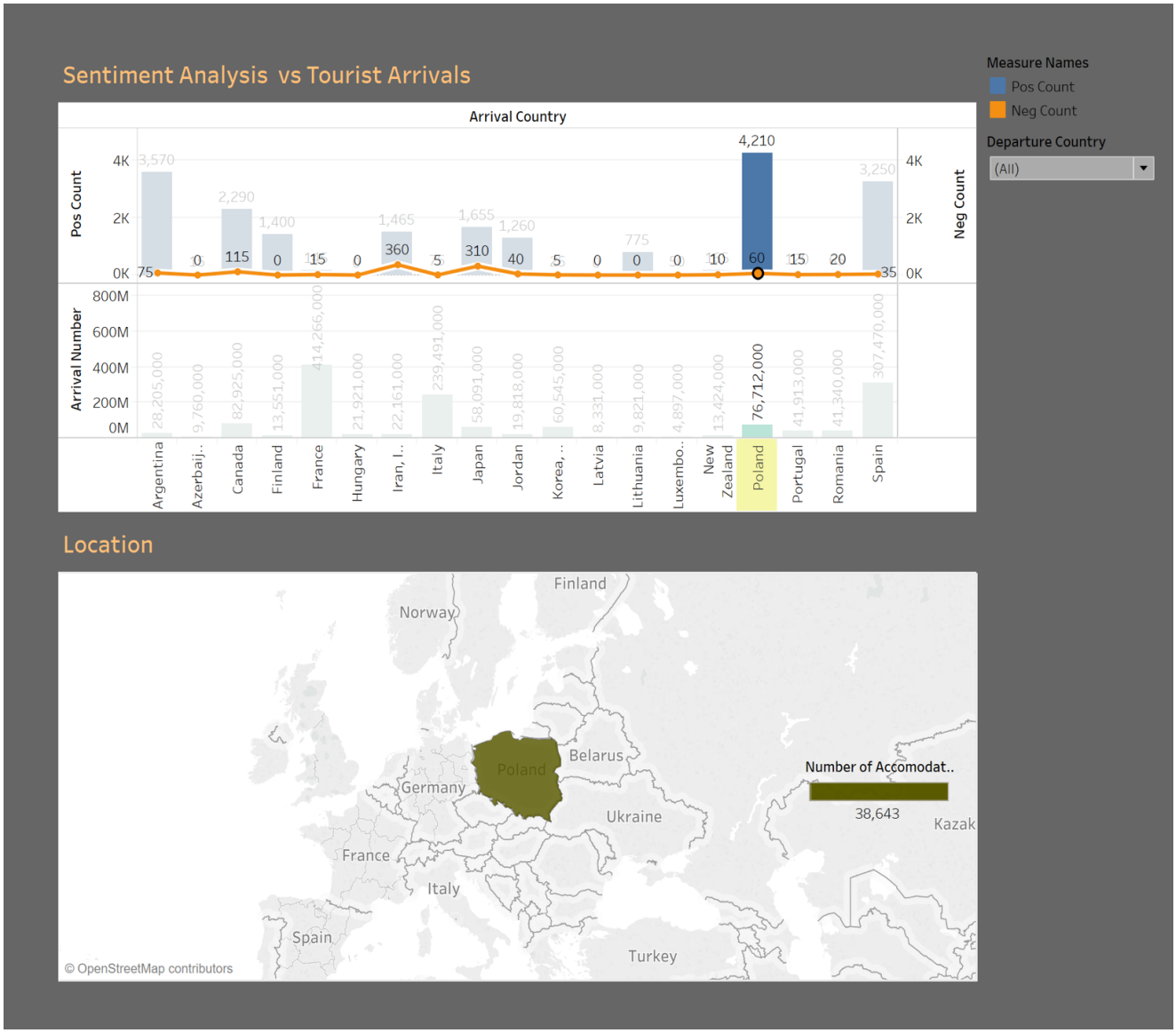

Figure 4

ANALYSIS OF THE OUTPUT

Answer to this query is the Sentiment Analysis plus the number of people going to a particular country and the number of accommodations (Dorms, Hostels, Hotels etc.) available for tourists. As we can see in the screenshot below that Poland has the maximum number of positive reviews from travelers & very few negative comments, the arrival number is seventy-six million. This shows that the people are finding enough accommodation and love to visit the country again and again. Similarly, the least negative count for France is ten which tells that it is the most attractive and enjoyable place around the globe which is justified by the number of arrivals in France i.e. 414 million. Therefore, we can conclude that Tourism in France is growing at good pace and is good for the countries growth as the sentiments, thoughts of people can be an important factor for flourishing of tourism in any part of the world.

SOURCES UTILIZED: -

Data Source {4}: - Sentiment Analysis - Twitter
Data Source {1}: - Tourists Arrivals
Data source {3}: - Tourists Accommodation

2nd Business Intelligence Query

What are the top 10 Destinations in the world?

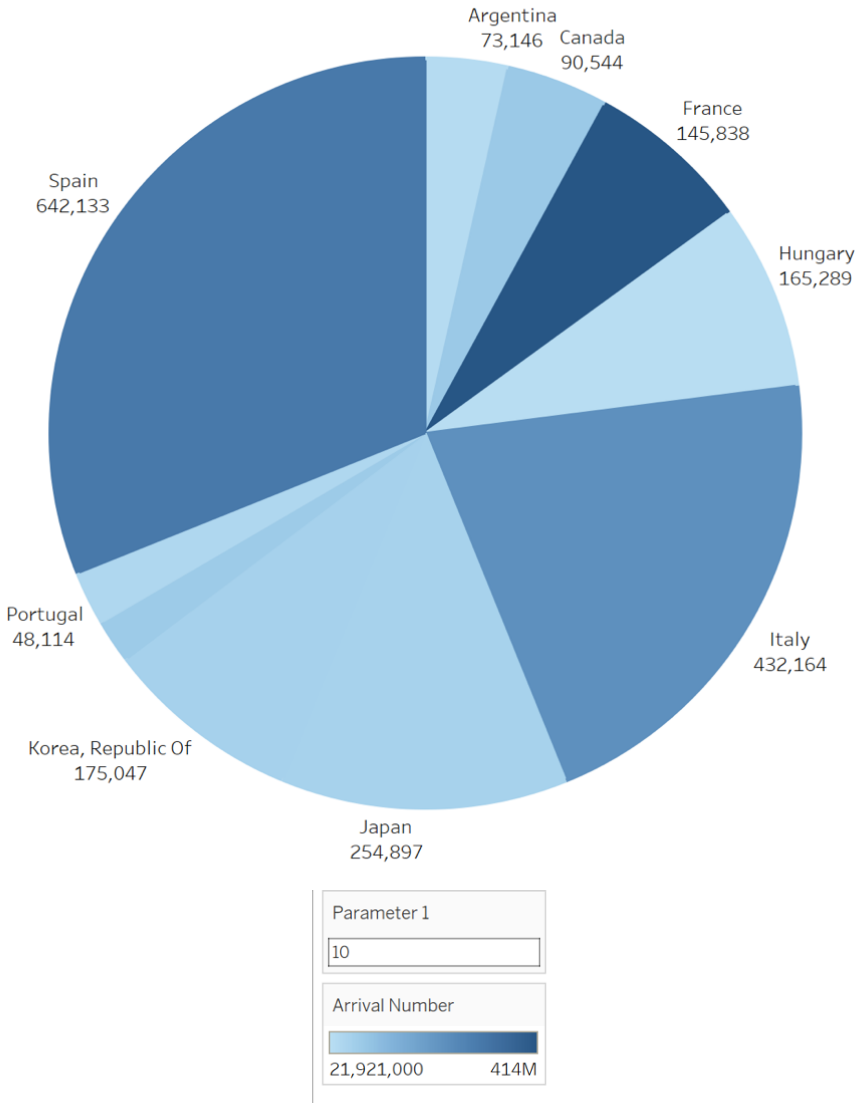

Figure 5

ANALYSIS OF THE OUTPUT

The top 10 destinations are given by comparing the number of arrivals of distinct country to the number of accommodations available for tourists. The screenshot shows the housing capacity of the individual countries and the parameters tell us the number of people going the particular country, that is shown by the range of the fading blue color. Dark blue color represents the country with highest number of tourists i.e. France with 414million travelers. The light blue color represents the country with lowest number of tourists based on the housing capacity i.e. Portugal. Thus, concluding that the number of accommodations available affects the people's choice of choosing a place for vacation.

SOURCES UTILIZED: -
Data Source {1}: - Tourists Arrivals
Data source {3}: - Tourists Accommodation

3rd Business Intelligence Query

What is the trend in tourism over the years for the top 3 Destinations?

Figure 6

ANALYSIS OF THE OUTPUT

- As we can see from the screenshot that tourism for France is expanding year by year and the percentage change for France is not big enough: - 5% change can be observed through the plot in the above screenshot but for the native French people from France the number has been fluctuating minutely, it is just 2% as shown in the above plot.
- For Spain, the percentage change for incoming tourists is: - 18% that is a significant change which is justified by my second query as is shows Spain has the largest number of accommodation available for tourists, it is increasing for the past 5 years gradually. The number of native Spanish people departing is reflected by a 10% change over the years as shown in the above plot.
- For Canada, the percentage change for incoming tourists is 11% and which is increasing year by year and is showing growth in the number of tourists. Whereas the percentage change for departing Canadians is 10% and there has been a dip in number over the years as shown in the above plot.

We conclude that the countries that are having an increase year by year are flooded by tourists and they should help widening the scope of tourism industry by encouraging various events and activities. The countries having more and more tourism per year should keep on improving their services and growth will be exponential as it helps in generating jobs and growth in economy. According to the trend more people are expected to go to France.

SOURCES UTILIZED:

Data source {1}: -Tourist Arrivals
Data Source {2}: -Tourist Departures

REFRENCING

[1] World Atlas: Complete list Of Country & Dialing Codes
Available: http://www.worldatlas.com/aatlas/ctycodes.htm [Accessed 20/11/17]
[2]. R. Kimball and M. Ross, The Data Warehouse: *The Definitive Guide to Dimensional Modeling* Toolkit, 3rd ed. Kimball Group, 2013

1) Code for Sentiment Analysis on Twitter using R:

```
library(devtools)
#install_github("geoffjentry/twitteR" , force=TRUE)
#install_github('R-package','quandl' , force=TRUE)
library(plyr)
library(httr)
library(doBy)
library(Quandl)
library(twitteR)
api_key <- "wHr7H5j3XZTe0RhrDzwIPBIKr"
api_secret <- "Mr0TDr1zmBfN9TUefXyIdMOTsPqpJm1sTecpMc5PShdtfWBUuM"
access_token <- "90846388-WaGrNPkuLYkoCUcqljJplAsuVg4pxBVm5wyQAhPsF"
access_token_secret <- "ib9kUo4xKWBLDKRrSRwe9uZ62OOpN12jl8Jlcs39BGDCu"
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
hu.liu.pos = scan('C:/Users/MOLAP/Desktop/positive-words.txt', what='character',comment.char=';')
hu.liu.neg = scan('C:/Users/MOLAP/Desktop/negative-words.txt', what='character',comment.char=';')
pos.words = c(hu.liu.pos, 'awesome','good','favorite','recommended','best')
neg.words = c(hu.liu.neg, 'wtf', 'boring', 'douzy', 'hilarious','terrible','fake','glorified','not recommended')
score.sentence <- 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)
}

score.sentiment <- function(sentences, pos.words, neg.words) {

  require(plyr)
  require(stringr)

  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    tryCatch(score.sentence(sentence, pos.words, neg.words ), error=function(e) 0)

  }, pos.words, neg.words)
```

```
  scores.df = data.frame(score=scores, text=sentences)

  return(scores.df)

}

collect.and.score <- function (handle,code,country,  pos.words, neg.words) {

  tweets = searchTwitter(handle, n=1500, lang="en", since=NULL, retryOnRateLimit=10)

  text = laply(tweets, function(t) t$getText())
  text <- sapply(text,function(row) iconv(row,"latin1","ASCII",sub = ""))
    score = score.sentiment(text, pos.words, neg.words)

  score$code = code

  score$country = country

  return (score)

}

India.scores = collect.and.score("@India","India",pos.words, neg.words)
Argentina.scores = collect.and.score("@Argentina","Argentina","ARG",pos.words, neg.words)
Albania.scores = collect.and.score("@Albania","Albania","ALB",pos.words, neg.words)
Austria.scores = collect.and.score("@Austria","Austria","AUT",pos.words, neg.words)
Azerbaijan.scores = collect.and.score("@Azerbaijan","Azerbaijan","AZE",pos.words, neg.words)
Belgium.scores = collect.and.score("@Belgium","Belgium","BEL",pos.words, neg.words)
Brazil.scores = collect.and.score("@Brazil","Brazil","BRA",pos.words, neg.words)
Bulgaria.scores = collect.and.score("@Bulgaria","Bulgaria","BGR",pos.words, neg.words)
Canada.scores = collect.and.score("@Canada","Canada","CAN",pos.words, neg.words)
CzechRepublic.scores = collect.and.score("@CzechRepublic","CzechRepublic","CZE",pos.words, neg.words)
Ecuador.scores = collect.and.score("@Ecuador","Ecuador","ECU",pos.words, neg.words)
Estonia.scores = collect.and.score("@Estonia","Estonia","EST",pos.words, neg.words)
Finland.scores = collect.and.score("@Finland","Finland","FIN",pos.words, neg.words)
France.scores = collect.and.score("@France","France","FRA",pos.words, neg.words)
Georgia.scores = collect.and.score("@Georgia","Georgia","GEO",pos.words, neg.words)
Germany.scores = collect.and.score("@Germany","Germany","DEU",pos.words, neg.words)
Hungary.scores = collect.and.score("@Hungary","Hungary","HUN",pos.words, neg.words)
Indonesia.scores = collect.and.score("@Indonesia","Indonesia","IDN",pos.words, neg.words)
Iran.scores = collect.and.score("@Iran","Iran","IRN",pos.words, neg.words)
Israel.scores = collect.and.score("@Israel","Israel","ISR",pos.words, neg.words)
Italy.scores = collect.and.score("@Italy","Italy","ITA",pos.words, neg.words)
Japan.scores = collect.and.score("@Japan","Japan","JPN",pos.words, neg.words)
Jordan.scores = collect.and.score("@Jordan"," Jordan","JOR",pos.words, neg.words)
Korea.scores = collect.and.score("@Korea","Korea","KOR",pos.words, neg.words)
Latvia.scores = collect.and.score("@Latvia","Latvia","LVA",pos.words, neg.words)
Lithuania.scores = collect.and.score("@Lithuania","Lithuania","LTU",pos.words, neg.words)
Luxembourg.scores = collect.and.score("@Luxembourg","Luxembourg","LUX",pos.words, neg.words)
NewZealand.scores = collect.and.score("@NewZealand","NewZealand","NZL",pos.words, neg.words)
Panama.scores = collect.and.score("@Panama","Panama","PAN",pos.words, neg.words)
Poland.scores = collect.and.score("@Poland","Poland","POL",pos.words, neg.words)
Portugal.scores = collect.and.score("@Portugal","Portugal","PRT",pos.words, neg.words)
```

```
Romania.scores = collect.and.score("@Romania","Romania","ROU",pos.words, neg.words)
RussianFederation.scores =
collect.and.score("@RussianFederation","RussianFederation","RUS",pos.words, neg.words)
SouthAfrica.scores = collect.and.score("@SouthAfrica","SouthAfrica","ZAF",pos.words, neg.words)
Spain.scores = collect.and.score("@Spain","Spain","ESP",pos.words, neg.words)


all.scores= rbind(Argentina.scores,Albania.scores,Austria.scores,Azerbaijan.scores , Belgium.scores
,Brazil.scores ,Bulgaria.scores ,Canada.scores, Ecuador.scores, Estonia.scores
                ,Finland.scores ,France.scores ,Georgia.scores,Germany.scores,Hungary.scores
,Indonesia.scores, Iran.scores ,Israel.scores, Italy.scores ,Japan.scores ,Jordan.scores, Korea.scores
,Latvia.scores
                ,Lithuania.scores,Luxembourg.scores,NewZealand.scores,Panama.scores
,Poland.scores,Portugal.scores ,Romania.scores,SouthAfrica.scores, Spain.scores
)

all.scores$very.pos = as.numeric( all.scores$score >= 2)

all.scores$very.neg = as.numeric( all.scores$score <= -2)

View(all.scores)

#near 0
all.scores$very.pos = as.numeric( all.scores$score >= 1)

all.scores$score = as.numeric( all.scores$score <= -1)

#the pos/neg sentiment scores for each airline

twitter.df = ddply(all.scores,c('code', 'country'), summarise, pos.count = sum (very.pos), neg.count =
sum(very.neg))

twitter.df$all.count = twitter.df$pos.count + twitter.df$neg.count

#sentiment score to be a percentage

twitter.df$score = round (100 * twitter.df$pos.count / twitter.df$all.count)

orderBy(~-score, twitter.df)

View(orderBy(~-score, twitter.df))

write.csv(orderBy(~-score, twitter.df),file='Sentiment.csv')
```

2.) SQL Query Code for Source table required while populating the cube.

```
SELECT * FROM [WorldTourism].[dbo].[RawDataArrivals] t
RIGHT JOIN [WorldTourism].[dbo].[RawDataAccommodation] a on a.Accommodation_ID=t.Arrival_ID
FULL OUTER JOIN [WorldTourism].[dbo].[RawDataTwitter] b on b.Sentiment_Code=t.Arrival_Code
LEFT JOIN [WorldTourism].[dbo].[RawDataDepartures] c on c.Departure_ID=t.Arrival_ID
```