

BIG DATA

PROYECTO FINAL

Jorge González Piedra



Jorge González Piedra

RESPONSABLE DEL DESARROLLO DE LA APLICACIÓN

Fecha: 05, FEB, 2023



DESARROLLO

1. Un informe científico, en el que se transmitan los resultados de los análisis realizados. Aquí explicaremos paso a paso cada uno de los apartados con las conclusiones correspondientes de las tareas realizadas. Podremos incluir secciones de código si es necesario y por supuesto, los resultados de cada una de las tareas realizadas sobre los datos obtenidos a través de la ejecución del código contenido en el documento técnico.
2. Un documento técnico que tendrá el código fuente (PySpark) empleado para la resolución de cada una de las tareas. El código fuente debe ser insertado como imágenes y con un tamaño que permita leer el texto contenido en las imágenes.
3. Una presentación guardada en formato pdf. Esta presentación nos servirá para mostrar los resultados de cada una de las tareas y no contendrá código fuente, sino que mostrará los resultados obtenidos siguiendo las guías de presentación que hemos visto en el módulo de proyectos *big data* y *storytelling*.



1. Fuente de datos.

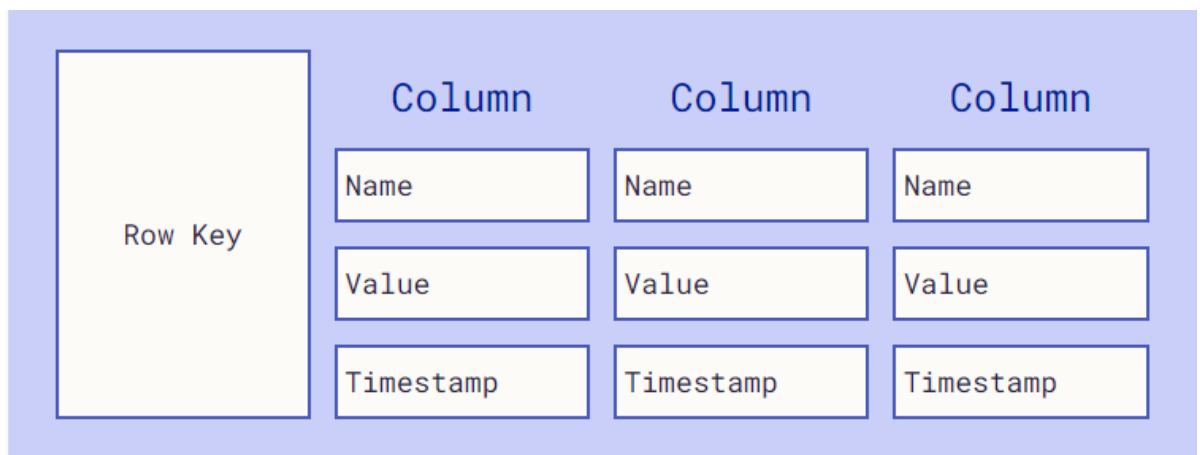
Analizamos la fuente de datos, **Air_Traffic_Passenger_Statistics.csv**.

Nombre del campo	Tipo de dato
Activity Period	integer (nullable = true)
Operating Airline	string (nullable = true)
Operating Airline IATA Code	string (nullable = true)
Published Airline	string (nullable = true)
Published Airline IATA Code	string (nullable = true)
GEO Summary	string (nullable = true)
GEO Region	string (nullable = true)
Activity Type Code	string (nullable = true)
Price Category Code	string (nullable = true)
Terminal	string (nullable = true)
Boarding Area	string (nullable = true)
Passenger Count	integer (nullable = true)
Adjusted Activity Type Code	string (nullable = true)
Adjusted Passenger Count	integer (nullable = true)
Year	integer (nullable = true)
Month	string (nullable = true)

2. Almacenamiento de datos.

Para el almacenamiento de los datos utilizaremos la base de datos distribuida NoSQL de tipo columnar **Cassandra**.

Las bases de datos orientadas a familias de columnas son similares a las bases de datos relacionales. La principal diferencia radica en que, a diferencia de las relacionales, que almacenan los datos en filas, en este tipo de bases de datos la información se almacena en columnas.



Mientras una base de datos relacional está optimizada para almacenar filas de datos, normalmente para aplicaciones transaccionales, una base de datos en columnas está optimizada para lograr una recuperación rápida de columnas de datos, normalmente en aplicaciones analíticas.

De la misma forma que otras bases de datos NoSQL, las bases de datos columnares están diseñadas para reducir la escala utilizando clústeres distribuidos

2.1. Creación de la base de datos.

Para trabajar con Cassandra utilizaremos DataStax.

DataStax es una infraestructura de datos escalable que permite manejar cualquier carga de trabajo en cualquier Cloud. Nos permite crear bases de datos de Apache Cassandra y conectarlas con aplicaciones externas mediante Apis o Drivers.

Se creará una nueva base de datos denominada **AirTrafficProyect**. Esta base de datos utilizará **GoogleCloud** para el almacenamiento de datos en la nube.




Create a Database

1 Enter the Basic Details

Database Name	Keyspace Name ⓘ
<input type="text" value="AirTrafficProyect"/>	<input type="text" value="airtrafic"/>
<small>Give it a memorable name - this can't be changed later.</small>	<small>Want to know more? Read our Docs to learn more about keyspaces.</small>

2 Select a Provider and Region

You've got access to 3 free regions in GCP. Unlock all regions by [upgrading to the Pay as you go plan](#).

 Google Cloud	 Amazon Web Services	 Microsoft Azure	Unlock All Regions
Select an Area		Select a Region	
North America 0 of 7 regions selected		<input checked="" type="radio"/> St. Ghislain, Belgium europe-west1	
Europe, Middle East, and Africa 1 of 2 regions selected >		<input type="radio"/> West Europe2 (London, England,... europe-... 🏠	
Asia Pacific 0 of 3 regions selected			
<small>Don't see the region you're after? Request a new region.</small>			

2.2. Creación de tablas.

Crearemos las tablas necesarias a partir de la fuente de datos. En nuestro caso crearemos la tabla airtraffic:

Tabla	Nombre del campo	Tipo de dato	Campos Primary Key	Campos Clustering Key
airtraffic	activity_period	integer (nullable = true)	Activity Period	Operating Airline
	operating_airline	string (nullable = true)		
	operating_airline_IAT A_code	string (nullable = true)		
	published_airline	string (nullable = true)		
	published_airline_IAT A_code	string (nullable = true)		
	GEO_summary	string (nullable = true)		
	GEO_region	string (nullable = true)		
	activity_type_code	string (nullable = true)		
	price_category_code	string (nullable = true)		
	terminal	string (nullable = true)		
	boarding_area	string (nullable = true)		

	passenger_count	integer (nullable = true)		
	adjusted_activity_type_code	string (nullable = true)		
	adjusted_passenger_count	integer (nullable = true)		
	year	integer (nullable = true)		
	month	string (nullable = true)		

Script de creación de la tabla:

```
CREATE TABLE IF NOT EXISTS airtrafic.airtraffic_table (  
  activity_period int,  
  operating_airline text,  
  operating_airline_IATA_code text,  
  published_airline text,  
  published_airline_IATA_code text,  
  GEO_summary text,  
  GEO_region text,  
  activity_type_code text,  
  price_category_code text,  
  terminal text,  
  boarding_area text,  
  passenger_count int,  
  adjusted_activity_type_code text,  
  adjusted_passenger_count int,  
  year int,  
  month text,  
  PRIMARY KEY(activity_period, operating_airline)  
)WITH CLUSTERING ORDER BY (operating_airline ASC);
```


2.3. Carga de datos.

Utilizaremos DataStax Bulk Loader que nos permitirá, a través del comando **dsbulk**, cargar todos los datos del CSV a nuestra base de datos creada en DataStax.
(La instalación de DSBULK se encuentra explicada en el ANEXO 1)

```
dsbulk load --connector.name csv --connector.csv.url Air_Traffic_Passen-  
ger_Statistics.csv -k airtrafic -t airtraffic_table -b "secure-connect-  
airtrafficproject.zip" -u stKCxJURhGhCMIjKHZYKFTLE -p hJbpsZq6tG0k0-  
KRl4IjPxM2Z98N-MIXAz.O..GRY3vcn+uMYeyt_WbNZYZ3HA-  
j75_v7J.b7vhXDXe0.WR3INmnN3cH_ZARa7xp,9oj,THzAMnOs+fYY29y7RND,2RN -header  
false -delim "," --schema.allowMissingFields true -m "0=activity_period,  
1=operating_airline, 2=operating_airline_iata_code, 3=published_airline ,  
4=published_airline_iata_code, 5=geo_summary ,6=geo_region, 7=activ-  
ity_type_code, 8=price_category_code, 9=terminal , 10=boarding_area ,  
11=passenger_count,12=adjusted_activity_type_code, 13=adjusted passen-  
ger_count, 14=year, 15=month"
```

```
Operation directory: C:\dsbulk-1.9.0\bin\logs\LOAD_20221223-071722-569000  
total | failed | rows/s | p50ms | p99ms | p999ms | batches  
15.007 | 0 | 16.843 | 98,12 | 126,35 | 127,40 | 24,68  
Operation LOAD_20221223-071722-569000 completed successfully in less than one second.  
Last processed positions can be found in positions.txt
```

2.4. Análisis de datos en Cassandra.

Para comprobar la correcta inserción de los datos llevaremos a cabo las siguientes consultas:

- Recuperar todos los registros de la aerolínea "Air China".

```
token@cqlsh:airtrafic> select * from airtrafic.airtraffic_table WHERE Op-  
erating_Airline = 'Air China' ALLOW FILTERING;
```

Utilizaremos DataStax Bulk para la extracción de los registros:

```
dsbulk unload -url airchina_data.csv -query "select activity_period ,op-  
erating_airline ,operating_airline_IATA_code ,published_airline ,pub-  
lished_airline_IATA_code ,GEO_summary ,GEO_region ,activity_type_code  
,price_category_code ,terminal ,boarding_area ,passenger_count ,ad-  
justed_activity_type_code,year,month from airtrafic.airtraffic_table  
WHERE Operating_Airline = 'Air China' ALLOW FILTERING" -b "secure-con-  
nect-airtrafficproyect.zip" -u stKCxJURhGhCMIjKHZYKFTLE -p hJbpsZq6tG0k0-  
KR14IjPxM2Z98N-MIXAz.O..GRY3vcn+uMYeyt_WbNZYZ3HA-  
j75_v7J.b7vhXDxeO.WR3INmnN3cH_ZARa7xp,9oj,THzAMnOs+fYY29y7RND,2RN
```

```
total | failed | rows/s | p50ms | p99ms | p999ms  
129 | 0 | 287 | 97,78 | 98,04 | 98,04  
Operation UNLOAD_20221223-152327-384000 completed successfully in less than one second.
```

Los datos recuperados se encuentran en el ANEXO 2

- Recuperar todos los vuelos de la compañía “Air Berlín” embarcados por la puerta “G”.

```
token@cqlsh:airtrafic> select * from airtrafic.airtraffic_table WHERE Op-  
erating_Airline = 'Air Berlin' AND boarding_area = 'G' ALLOW FILTERING;
```

Utilizaremos DataStax Bulk para la extracción de los registros:

```
dsbulk unload -url airberlin_data.csv -query "select activity_period ,op-  
erating_airline ,operating_airline_IATA_code ,published_airline ,pub-  
lished_airline_IATA_code ,GEO_summary ,GEO_region ,activity_type_code  
,price_category_code ,terminal ,boarding_area ,passenger_count ,ad-  
justed_activity_type_code,year,month from airtrafic.airtraffic_table  
WHERE Operating_Airline = 'Air Berlin' AND boarding_area = 'G' ALLOW FIL-  
TERING" -b "secure-connect-airtrafficproyect.zip" -u stKCxJURhGh-  
CMIjKHZYKFTLE -p hJbpsZq6tG0k0-KR14IjPxM2Z98N-  
MIXAz.O..GRY3vcn+uMYeyt_WbNZYZ3HA-  
j75_v7J.b7vhXDxeO.WR3INmnN3cH_ZARa7xp,9oj,THzAMnOs+fYY29y7RND,2RN
```

```
Operation directory: C:\dsbulk-1.9.0\bin\logs\UNLOAD_20221224-093335-090000  
total | failed | rows/s | p50ms | p99ms | p999ms  
6 | 0 | 20 | 41,29 | 41,42 | 41,42  
Operation UNLOAD_20221224-093335-090000 completed successfully in less than one second.
```

Los datos recuperados se encuentran en el ANEXO 3

3. Análisis preliminar de datos.

Crearemos un Notebook llamado AirTraffic_Jorge_Gonzalez_Piedra que se adjuntará a la entrega.

3.1.Cargar datos en DataFrame.

Crearemos un DataFrame donde se cargarán los datos del csv *Air_Traffic_Passenger_Statistics.csv*

```
from pyspark.sql.functions import col

df_airport = spark.read.options(inferSchema='True',delimiter=',',
header=True).csv("/content/drive/MyDrive/TOKIO/Big Data - Cloud Compu-
ting/01 - Big Data/PROYECTO FINAL/Air_Traffic_Passenger_Statistics.csv")
```

```
df_airport.printSchema()
```

```
root
 |-- Activity Period: integer (nullable = true)
 |-- Operating Airline: string (nullable = true)
 |-- Operating Airline IATA Code: string (nullable = true)
 |-- Published Airline: string (nullable = true)
 |-- Published Airline IATA Code: string (nullable = true)
 |-- GEO Summary: string (nullable = true)
 |-- GEO Region: string (nullable = true)
 |-- Activity Type Code: string (nullable = true)
 |-- Price Category Code: string (nullable = true)
 |-- Terminal: string (nullable = true)
 |-- Boarding Area: string (nullable = true)
 |-- Passenger Count: integer (nullable = true)
 |-- Adjusted Activity Type Code: string (nullable = true)
 |-- Adjusted Passenger Count: integer (nullable = true)
 |-- Year: integer (nullable = true)
 |-- Month: string (nullable = true)
```

Para evitar problemas renombramos las columnas eliminando los espacios en blanco:

```
df_airport = df_airport.withColumnRenamed("Activity Period", "ActivityPeriod") \
.withColumnRenamed("Activity Period", "ActivityPeriod") \
.withColumnRenamed("Operating Airline", "OperatinAirline") \
.withColumnRenamed("Operating Airline IATA Code", "OperatingAirlineIATACode") \
.withColumnRenamed("Published Airline", "PublishedAirline") \
.withColumnRenamed("Published Airline IATA Code", "PublishedAirlineIATACode") \
.withColumnRenamed("GEO Summary", "GEOSummary") \
.withColumnRenamed("GEO Region", "GEORegion") \
.withColumnRenamed("Activity Type Code", "ActivityTypeCode") \
.withColumnRenamed("Price Category Code", "PriceCategoryCode") \
.withColumnRenamed("Boarding Area", "BoardingArea") \
.withColumnRenamed("Passenger Count", "PassengerCount") \
.withColumnRenamed("Adjusted Activity Type Code", "AdjustedActivityTypeCode") \
.withColumnRenamed("Adjusted Passenger Count", "AdjustedPassengerCount")
```

```
root
|-- ActivityPeriod: integer (nullable = true)
|-- OperatinAirline: string (nullable = true)
|-- OperatingAirlineIATACode: string (nullable = true)
|-- PublishedAirline: string (nullable = true)
|-- PublishedAirlineIATACode: string (nullable = true)
|-- GEOSummary: string (nullable = true)
|-- GEORegion: string (nullable = true)
|-- ActivityTypeCode: string (nullable = true)
|-- PriceCategoryCode: string (nullable = true)
|-- Terminal: string (nullable = true)
|-- BoardingArea: string (nullable = true)
|-- PassengerCount: integer (nullable = true)
|-- AdjustedActivityTypeCode: string (nullable = true)
|-- AdjustedPassengerCount: integer (nullable = true)
|-- Year: integer (nullable = true)
|-- Month: string (nullable = true)
```

3.2. Análisis de datos.

Analizaremos los datos del CSV respondiendo las siguientes preguntas:

- **¿Cuántas compañías diferentes aparecen en el fichero?**

Existen 77 compañías diferentes.

```
df_airport.dropDuplicates(["OperatingAirline"]).select("OperatingAir-  
line").count()  
df_airport.dropDuplicates(["OperatingAirline"]).select("OperatingAir-  
line").show()
```

```
df_airport.dropDuplicates(["OperatingAirline"]).select("OperatingAirline").count()  
77
```

Los datos recuperados se encuentran en el ANEXO 4

- **¿Cuántos pasajeros tienen de media los vuelos de cada compañía?**

Mostraremos la media de los campos PassengerCount y AdjustedPassengerCount:

```
df_airport.groupBy("OperatingAirline").mean("PassengerCount", "Adjusted-  
PassengerCount").show()
```

Los datos recuperados se encuentran en el ANEXO 5.

3.3. Eliminación de registros duplicados por el campo GEO Region.

Eliminaremos los registros duplicados por el campo GEORegion manteniendo solo el registro con mayor número de pasajeros (campo PassengerCount).

Para ello crearemos una vista temporal sobre el DataFrame y mediante una consulta SQL agruparemos por GEORegion recuperando el registro cuyo valor de PassengerCount sea más alto:

```
df_GEORegion_no_duplicates = spark.sql("select a1.* FROM " \
"df_airport_view a1, " \
"(SELECT GEORegion, MAX(PassengerCount) PassengerCount FROM df_air-
port_view GROUP BY GEORegion) a2 " \
"WHERE a1.GEORegion = a2.GEORegion " \
"AND a1.PassengerCount = a2.PassengerCount")
df_GEORegion_no_duplicates.show();
```

3.4. Volcar nuevos datos a CSV.

Se volcarán los datos del punto anterior a un CSV llamado *airtraffic_drop_duplicates_georegion.csv*. Se adjuntará dicho CSV a la entrega:

```
df_GEORegion_no_duplicates.write.options(header="True").csv("/con-
tent/drive/MyDrive/TOKIO/Big Data - Cloud Computing/01 - Big Data/PROY-
ECTO FINAL/Entrega/Ficheros/airtraffic_drop_duplicates_georegion")
```

Los datos extraídos se encuentran en el ANEXO 6.



4. Análisis estadístico

4.1. Análisis descriptivo

Estudiaremos las variables descriptivas de cada uno de los datos (media, moda, desviación estándar, mínimo y máximo).

Para los datos categóricos estudiaremos solo la moda.

Para obtener las variables descriptivas utilizaremos la función **describe()** de pyspark, para la obtención de la moda, convertiremos el DataFrame a un DataFrame de pandas y utilizaremos la función **mode()**.

4.1.1. Activity period

Dato categórico que se refiere a un periodo de tiempo mes-año. Por ejemplo, el periodo de actividad 200507 se refiere a Julio de 2005, el 200508 a Agosto de 2005 y así sucesivamente. El valor mínimo, es decir, el primer periodo estudiado es 200507 (Julio 2005) y el valor máximo, el último periodo estudiado es 201603 (Marzo 2016).

A pesar de ser un dato categórico, utilizaremos el comando **describe()**, pero solo nos fijaremos en los valores máximo y mínimo para entender los periodos que estudiaremos.

```
[21] df_airport.describe("ActivityPeriod").show()
```

```
+-----+-----+
|summary| ActivityPeriod|
+-----+-----+
|  count|           15007|
|   mean| 201045.07336576266|
|  stddev| 313.33619609986414|
|    min|           200507|
|    max|           201603|
+-----+-----+
```

```
[41] df_airport_pandas['ActivityPeriod'].mode()
```

```
0    200807
dtype: int32
```

Observamos que el valor más repetido es 200807. Teniendo en cuenta que cada registro de nuestro DataFrame representa un vuelo, podemos afirmar que en este periodo que se corresponde con Julio de 2008, fue el periodo con más vuelos en nuestro aeropuerto.

4.1.2. Operating airline

Dato que representa el nombre de la aerolínea que realiza cada vuelo.

```
df_airport_pandas['OperatingAirline'].mode()
0    United Airlines - Pre 07/01/2013
dtype: object
```

Observamos que el valor más repetido, es decir, la compañía que más vuelo realiza es United Airlines – Pre 07/01/2013.


```
df_airport.filter(df_airport.OperatingAirline == "United Airlines - Pre 07/01/2013").dropDuplicates().count()
```

2154

Observamos que la compañía ha realizado 2154 vuelos.

4.1.3. Operating airline IATA code

Dato que representa el código de la compañía en IATA (International Air Transport Association).

```
df_airport_pandas['OperatingAirlineIATACode'].mode()
```

0 UA
dtype: object

El valor más repetido es código asociado a la compañía aérea *United Airlines – Pre 07/01/2013* que hemos visto en el punto anterior.

4.1.4. Published airline

Dato que representa el código de la compañía en IATA (International Air Transport Association).

```
df_airport_pandas['PublishedAirline'].mode()
```

0 United Airlines - Pre 07/01/2013
dtype: object

La más repetida, de nuevo, es la compañía *United Airlines – Pre 07/01/2013*

4.1.5. Published airline IATA code

Dato que representa el código de la compañía publicado en IATA (International Air Transport Association).

```
df_airport_pandas['PublishedAirlineIATACode'].mode()

0    UA
dtype: object
```

El más repetido, de nuevo, es la asociada con la compañía United Airlines – Pre 07/01/2013.

4.1.6. GEO summary

Dato que representa el tipo de vuelo. Los valores que puede tomar son International o Domestic:

```
[66] df_airport.select("GEOSummary").dropDuplicates().show()
```

```
+-----+
| GEOSummary |
+-----+
| International |
| Domestic |
+-----+
```

```
df_airport_pandas['GEOSummary'].mode()

0    International
dtype: object
```

La moda indica los vuelos suelen ser en mayor parte Internacionales.

4.1.7. GEO region

Dato que representa la región de destino de nuestro vuelo.

```
df_airport_pandas['GEORegion'].mode()
0    US
dtype: object
```

En nuestros datos aparecen 9 regiones posibles:

```
df_airport.select("GEORegion").dropDuplicates().show()
+-----+
|      GEORegion|
+-----+
|      Europe|
| Central America|
|         US|
| South America|
|      Mexico|
|   Middle East|
|      Canada|
|Australia / Oceania|
|         Asia|
+-----+
```

El valor más repetido es US, es decir, la mayoría de los vuelos que pasan por nuestro aeropuerto tienen como región de destino final Estados Unidos.

4.1.8. Activity type code

Dato que representa el tipo de actividad del vuelo. Los valores posibles son, *Enplaned* (Planificado), *Thru/Transit* (En tránsito), *Deplaned* (Desplanificado).

```
df_airport.select("ActivityTypeCode").dropDuplicates().show()
```

ActivityTypeCode
Enplaned
Thru / Transit
Deplaned

```
df_airport_pandas['ActivityTypeCode'].mode()
```

```
0    Deplaned
dtype: object
```

Observamos que el valor más repetido es *Deplaned*, es decir la mayoría de vuelos que estudiaremos serán vuelos desplanificados.



4.1.9. Price category code

Dato que representa el tipo de tarifa del vuelo. Los posibles valores son *Low Fare* y *Other*. Suponemos que *Other* son tarifas medias y altas y *Low Fare* tarifas bajas.

```
df_airport.select("PriceCategoryCode").dropDuplicates().show()
```

```
+-----+  
|PriceCategoryCode|  
+-----+  
|                |Other|  
|                |Low Fare|  
+-----+
```

```
df_airport_pandas['PriceCategoryCode'].mode()
```

```
0    Other
dtype: object
```

Vemos que el valor más repetido es *Other*. Por lo que la mayor parte de los vuelos, podemos suponer que son tarifas medias y altas.



4.1.10. Terminal

Dato que representa la terminal del vuelo. Los valores posibles son:

```
df_airport.select("Terminal").dropDuplicates().show()
```

```
+-----+  
|   Terminal|  
+-----+  
|International|  
|      Other|  
| Terminal 3|  
| Terminal 2|  
| Terminal 1|  
+-----+
```

```
df_airport_pandas['Terminal'].mode()
```

```
0    International  
dtype: object
```

Observamos que el valor más repetido es International. Este dato se corresponde con GEO Summary donde el valor más repetido era también International.



4.1.11. Boarding area

Dato que representa la terminal del embarque del vuelo. Los valores posibles son:

```
df_airport.select("BoardingArea").dropDuplicates().show()
```

BoardingArea
F
E
B
D
Other
C
A
G

```
df_airport_pandas['BoardingArea'].mode()
```

```
0      A
dtype: object
```

Observamos que la mayoría de vuelos embarcan por la zona A.

4.1.12. Passenger count

Dato que representa el número de pasajeros de cada vuelo.

```
df_airport.describe("PassengerCount").show()
```

```
summary PassengerCount
count          15007
mean  29240.521090157927
stddev  58319.509284123524
min           1
max        659837
```

Observamos que, de media, cada vuelo lleva a 29240.52 pasajeros.

4.1.13. Adjusted activity type code

Dato que representa el tipo de actividad, ajustado, suponemos, para los datos nulos o vacíos. De ahora en adelante, si fuera necesario, se utilizará este campo, y no Activity type code, para los cálculos.

```
df_airport_pandas['AdjustedActivityTypeCode'].mode()
```

```
0    Deplaned
dtype: object
```

Observamos el dato más repetido es *Desplanificado* (Deplaned), coincidiendo con la moda del campo Activity Type Code.

4.1.14. Adjusted passenger count

Dato que representa el número de pasajeros, ajustado, suponemos, para los datos nulos o vacíos. De ahora en adelante, si fuera necesario, se utilizará este campo y no Passenger count para los cálculos.

```
df_airport.describe("AdjustedPassengerCount").show()
```

```
| summary | AdjustedPassengerCount |
+-----+-----+
| count | 15007 |
| mean | 29331.917105350836 |
| stddev | 58284.1822186625 |
| min | 1 |
| max | 659837 |
+-----+-----+
```

Observamos que la media es muy cercana a la media del dato Passenger count.

4.1.15. Year

Dato que representa el año del vuelo.

```
[62] df_airport.describe("Year").show()
```

```
| summary | Year |
+-----+-----+
| count | 15007 |
| mean | 2010.385220230559 |
| stddev | 3.137589043169972 |
| min | 2005 |
| max | 2016 |
+-----+-----+
```

```
[63] df_airport_pandas['Year'].mode()
```

```
0    2015
dtype: int32
```

Observamos que el año en el que más vuelos se realizaron fue 2015.



4.1.16.Month

Dato que representa el mes del vuelo.

```
df_airport_pandas['Month'].mode()

0    August
dtype: object
```

Observamos que el mes en el que más vuelos se realizan es Agosto. Esto puede ser debido al comienzo de las vacaciones de verano.

4.2. Análisis de correlación

Realizaremos un análisis de correlación de las variables de nuestro DataFrame. Para ello, convertiremos nuestro DataFrame a uno de Pandas y utilizaremos la función `corr()`, lo que nos devolverá una matriz de correlación de todas las columnas.

```
df_airport_pd.corr()
```

Para una mejor visualización de esta matriz, añadiremos un estilo que nos permita diferenciar por colores aquellas correlaciones más fuertes:

```
df_airport_pd.corr().style.background_gradient(cmap='coolwarm')
```

	ActivityPeriod	OperatingAirline	OperatingAirlineIATACode	PublishedAirline	PublishedAirlineIATACode	GEOSummary	GEORegion	ActivityTypeCode	PriceCategoryCode	Terminal	BoardingArea	PassengerCount	AdjustedActivityTypeCode	AdjustedPassengerCount	Year	Month
ActivityPeriod	1.000000	0.008132	-0.043771	0.009796	-0.010936	0.066247	-0.028159	-0.052887	-0.006257	-0.008901	-0.005209	0.061871	-0.052687	0.060089	0.995763	-0.002869
OperatingAirline	0.008132	1.000000	0.823688	0.968828	0.818090	-0.130956	0.151120	0.100644	-0.096112	0.197959	0.251975	0.185424	0.180644	0.186427	0.008183	-0.000452
OperatingAirlineIATACode	-0.043771	0.823688	1.000000	0.799621	0.919021	-0.131919	0.101535	0.099556	-0.091939	0.200314	0.280294	0.122244	0.099556	0.123170	-0.043257	0.000861
PublishedAirline	0.009796	0.968828	0.799621	1.000000	0.859995	-0.083020	0.108379	0.098414	-0.095264	0.199894	0.275990	0.200862	0.098414	0.201890	0.009641	0.000609
PublishedAirlineIATACode	-0.010936	0.818090	0.919021	0.859995	1.000000	-0.027591	0.008986	0.097818	-0.105385	0.167910	0.312936	0.155368	0.097818	0.156337	-0.010837	0.001689
GEOSummary	0.066247	-0.130956	-0.131919	-0.083020	-0.027591	1.000000	-0.871826	-0.026760	0.411498	-0.574422	0.109553	-0.395743	-0.026760	-0.396856	0.966046	-0.001139
GEORegion	-0.028159	0.151120	0.101535	0.108379	0.008986	-0.871826	1.000000	0.033899	-0.382864	0.509119	-0.121033	0.336113	0.033899	0.336880	-0.028129	0.009949
ActivityTypeCode	-0.052887	0.100644	0.099556	0.098414	0.097818	-0.026760	0.033899	1.000000	0.001004	0.087788	0.087786	-0.071423	1.000000	-0.067804	-0.052364	-0.001523
PriceCategoryCode	-0.006257	-0.096112	-0.091939	-0.095264	-0.105385	0.411498	-0.382864	0.001004	1.000000	-0.102936	0.213485	-0.065047	0.001004	-0.064681	-0.005683	-0.003627
Terminal	-0.088901	0.197959	0.200314	0.199894	0.167910	-0.574422	0.509119	0.087788	-0.102936	1.000000	0.168414	0.429146	0.087788	0.430687	-0.088155	-0.000093
BoardingArea	-0.005209	0.251975	0.280294	0.275990	0.312936	0.109553	-0.121033	0.087786	0.213485	0.168414	1.000000	0.131091	0.087786	0.132147	-0.005109	-0.000581
PassengerCount	0.061871	0.185424	0.122244	0.200862	0.155368	-0.395743	0.336113	-0.071423	0.065047	0.429146	0.131091	1.000000	-0.071423	0.999941	0.060089	0.000413
AdjustedActivityTypeCode	-0.052887	0.100644	0.099556	0.098414	0.097818	-0.026760	0.033899	1.000000	0.001004	0.087788	0.087786	-0.071423	1.000000	-0.067804	-0.052364	-0.001523
AdjustedPassengerCount	0.060089	0.186427	0.123170	0.201890	0.156337	-0.396856	0.336880	-0.067804	-0.064681	0.430687	0.132147	0.999941	-0.067804	1.000000	0.059096	0.000365
Year	0.995763	0.008183	-0.043257	0.009641	-0.010837	0.966046	-0.028129	-0.052364	-0.005683	-0.088155	-0.005109	0.060089	-0.052364	0.059096	1.000000	0.030413
Month	-0.002869	-0.000452	0.000861	0.000609	0.001689	-0.001139	0.009949	-0.001523	-0.003627	-0.000093	-0.000581	0.000413	-0.001523	0.000365	-0.030413	1.000000

Esta matriz se encuentra adjunta a la entrega en la carpeta `airtraffic_correlation_matrix`.

Es importante recalcar que, aunque hemos indexado las variables categóricas y hemos sacado una correlación que estudiaremos a continuación (ya que es técnicamente posible) no tiene por qué haber necesariamente una relación directa entre variables categóricas, aunque la matriz la muestre.

Nota: Por ejemplo, en la correlación `OperatingAirline` – `GEOSummary` podemos indexar ambas columnas de manera que para `Operating Airline` la compañía “Air Canada” sea 1 y “Virgin America” sea 2 y para `GEOSummary` el valor “Domestic” sea 1 e “International” 2. Si `Operating Airline` aumenta al mismo tiempo que `GeoSummary` esto no tiene que significar realmente que haya una correlación porque ambos son datos categóricos. En nuestro caso el valor 2 representa a “Virgin

America", no es un dato numérico que represente un incremento del valor de la variable, simplemente una categoría distinta.

Mostramos a continuación las correlaciones más fuertes de esta matriz:

4.2.1.GEO Summary – Price Category Code

```
[ ] df_airport_pd.GEOSummary.corr(df_airport_pd.PriceCategoryCode)

0.41149848056451377
```

4.2.2.GEO Summary – GEO Region

```
[ ] df_airport_pd.GEOSummary.corr(df_airport_pd.GEORegion)

-0.8718261857198394
```

4.2.3.GEO Region – Terminal

```
[ ] df_airport_pd.GEORegion.corr(df_airport_pd.Terminal)

0.5091186306605863
```

4.2.4.GEO Region – Price Category Code

```
[ ] df_airport_pd.PriceCategoryCode.corr(df_airport_pd.GEORegion)

-0.3828639102138204
```

4.2.5.Adjusted Passenger Count – GEO Region

```
[ ] df_airport_pd.AdjustedPassengerCount.corr(df_airport_pd.GEORegion)

0.3369804846146507
```

4.2.6.Adjusted Passenger Count – Terminal

```
[ ] df_airport_pd.AdjustedPassengerCount.corr(df_airport_pd.Terminal)

0.43068707562529646
```



Para estudiar de manera más precisa la correlación entre las variables dicotómicas (aquellas variables categóricas que solo pueden tomar dos posibles valores) GEO Summary (*Domestic, International*) y Price category code (*Low fare, Other*) con el número de pasajeros, utilizaremos el **método Point-Biserial**.

4.2.7.GEO Summary– Adjusted passenger count

```
[58] stats.pointbiserialr(df_airport_pd['GEOSummary'], df_airport_pd['AdjustedPassengerCount'])  
  
PointbiserialrResult(correlation=-0.39685620097984975, pvalue=0.0)
```

4.2.8.Price category code – Adjusted passenger count

```
[59] stats.pointbiserialr(df_airport_pd['PriceCategoryCode'], df_airport_pd['AdjustedPassengerCount'])  
  
PointbiserialrResult(correlation=-0.0646612429860395, pvalue=2.2120528625642906e-15)
```

4.3. Regresión lineal

Realizaremos una regresión lineal estudiando la cantidad de pasajeros que pasan por nuestro aeropuerto a lo largo del tiempo para intentar predecir la cantidad de pasajeros que tendremos en los próximos años.

Comenzaremos creando un nuevo DataFrame para estudiar el número de pasajeros que hemos tenido a lo largo de los años. Este nuevo DataFrame contendrá las columnas *Year* y *Month*. Además, tendremos la columna *PassengerCountSum* que será el sumatorio de todos los pasajeros de todos los vuelos del año-mes:

```
from pyspark.sql.functions import sum
df_pass_by_year = df_airport.groupBy("Year" , "Month").agg(sum("Passen-
gerCount").alias("PassengerCountSum"))
df_pass_by_year.show()
```

Year	Month	PassengerCountSum
2006	September	2720100
2007	May	3056934
2012	February	2998119
2008	April	3029021
2006	October	2834959
2011	November	3326859
2006	February	2223024
2014	July	4499221
2011	August	3917884
2007	December	2903637
2013	August	4347059
2009	February	2359800
2014	May	4147096
2011	October	3602455
2006	July	3227605
2006	November	2653887
2014	November	3628786
2009	May	3177100
2013	December	3814984
2014	December	3855835

only showing top 20 rows

Convertiremos los valores de la columna *Month* a tipo numérico, de manera que January pase a ser 1, February 2 y así sucesivamente. Para ello convertiremos nuestro nuevo DataFrame a uno de Pandas y a través de **map()** y del módulo **calendar** realizaremos la conversión:

```
df_pass_by_year_pd = df_pass_by_year.toPandas()
```

```
import calendar as cal

lower_ma = [m.lower() for m in cal.month_name]
df_pass_by_year_pd['Month'] =
df_pass_by_year_pd['Month'].str.lower().map(lambda m: lower_ma.index(m)).astype('Int8')
```

 `df_pass_by_year_pd.sort_values(by=['Year', 'Month'])`

	Year	Month	PassengerCountSum
60	2005	7	3225769
62	2005	8	3195866
90	2005	9	2740553
117	2005	10	2770715
84	2005	11	2617333
...
109	2015	11	4013814
82	2015	12	4129052
113	2016	1	3748529
85	2016	2	3543639
31	2016	3	4137679

129 rows x 3 columns

Crearemos una nueva columna para unir Year y Month en un campo de tipo fecha:

```
df_pass_by_year_pd['Date'] = df_pass_by_year_pd[df_pass_by_year_pd.columns[0:2]].apply(lambda x: "-".join(x.values.astype(str)),axis="columns")
df_pass_by_year_pd['Date']=
pd.to_datetime(df_pass_by_year_pd['Date']).dt.strftime("%Y-%m")
df_pass_by_year_pd.sort_values(by=["Date"])
```

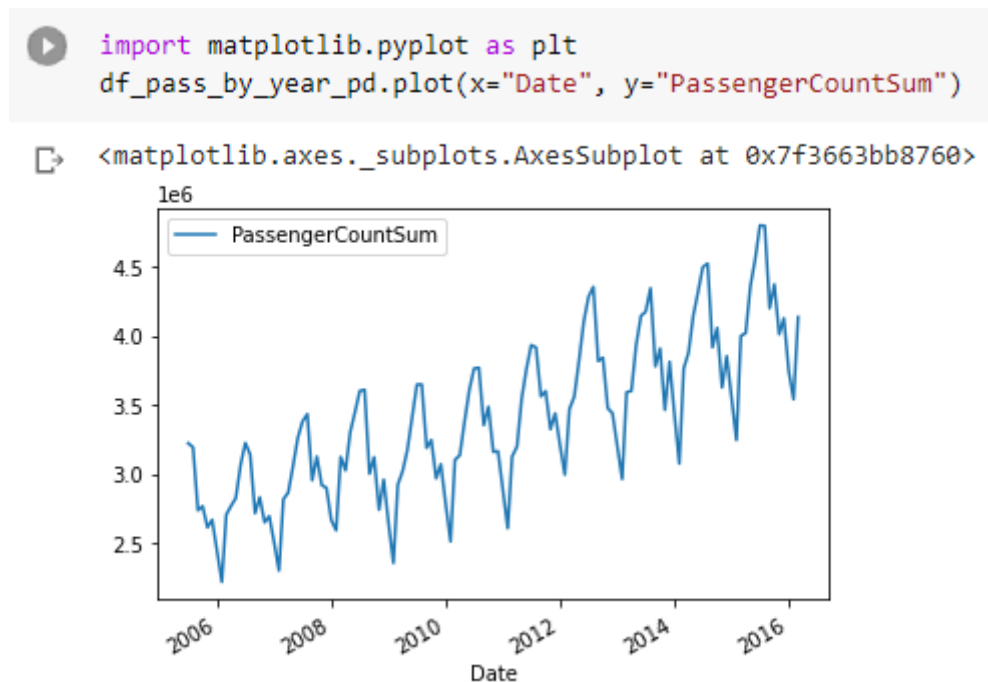
	Year	Month	PassengerCountSum	Date
60	2005	7	3225769	2005-07
62	2005	8	3195866	2005-08
90	2005	9	2740553	2005-09
117	2005	10	2770715	2005-10
84	2005	11	2617333	2005-11
...
109	2015	11	4013814	2015-11
82	2015	12	4129052	2015-12
113	2016	1	3748529	2016-01
85	2016	2	3543639	2016-02
31	2016	3	4137679	2016-03

129 rows x 4 columns

```
df_pass_by_year_pd.info()

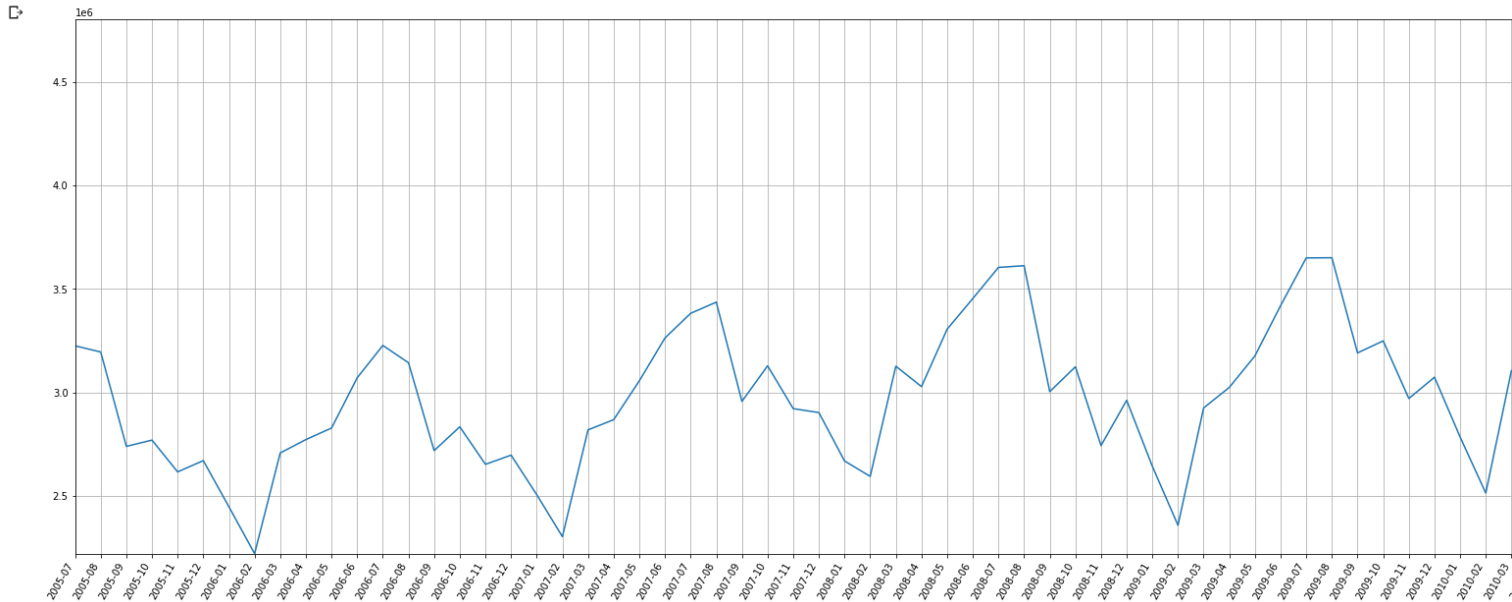
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129 entries, 0 to 128
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Year                  129 non-null   int32  
1   Month                 129 non-null   Int8    
2   PassengerCountSum     129 non-null   int64  
3   Date                  129 non-null   datetime64[ns]
dtypes: Int8(1), datetime64[ns](1), int32(1), int64(1)
memory usage: 2.9 KB
```


Utilizando la librería matplotlib dibujaremos estos datos en un gráfico para entender un poco mejor los datos:



Observamos un patrón recurrente. El número de pasajeros desciende, y asciende de manera periódica. Para estudiar esto un poco más en detalle hacemos zoom en el gráfico y añadimos un grid:

```
import matplotlib.pyplot as plt
graph_airtraffic = plt.plot(df_pass_by_year_pd['Date'], df_pass_by_year_pd['PassengerCountSum'], label="Datos reales")
plt.rcParams["figure.figsize"] = (60,10)
plt.grid(True)
plt.xticks(rotation=60,ha="right")
plt.margins(0)
```



Observamos que el mes con menos afluencia de pasajeros suele ser Febrero. Los meses con más afluencia de pasajeros son **Julio-Agosto**, coincidiendo con el verano y las vacaciones. A partir de Agosto comienza un descenso del número de pasajeros hasta volver a llegar a **Febrero** que comienza a ascender de nuevo.

Una vez visto este patrón y estudiado los datos vamos a crear la regresión.

Comenzamos importando el módulo de linearModel de sklearn:

```
from sklearn import linear_model
```

Para realizar la regresión linear no podemos utilizar el campo de tipo fecha por lo que convertiremos el campo Date a un ordinal para seguir manteniendo el orden de los datos:

```
import datetime as dt
df_pass_by_year_pd['Date'] = pd.to_datetime(df_pass_by_year_pd['Date'])
df_pass_by_year_pd['DateOrd'] = df_pass_by_year_pd['Date'].map(dt.datetime.
toordinal)
df_pass_by_year_pd = df_pass_by_year_pd.sort_values(by=["DateOrd"])
```

	Year	Month	PassengerCountSum	Date	DateOrd
60	2005	7	3225769	2005-07-01	732128
62	2005	8	3195866	2005-08-01	732159
90	2005	9	2740553	2005-09-01	732190
117	2005	10	2770715	2005-10-01	732220
84	2005	11	2617333	2005-11-01	732251
...
109	2015	11	4013814	2015-11-01	735903
82	2015	12	4129052	2015-12-01	735933
113	2016	1	3748529	2016-01-01	735964
85	2016	2	3543639	2016-02-01	735995
31	2016	3	4137679	2016-03-01	736024

Simplificamos el DataFrame para quedarnos con los datos que realmente estudiaremos, PassengerCountSum, Date y DateOrd:

```
df_pass_by_year_pd_simple = df_pass_by_year_pd[["PassengerCountSum", "Date", "DateOrd"]]  
df_pass_by_year_pd_simple
```

	PassengerCountSum	Date	DateOrd
60	3225769	2005-07	732128
62	3195866	2005-08	732159
90	2740553	2005-09	732190
117	2770715	2005-10	732220
84	2617333	2005-11	732251
...
109	4013814	2015-11	735903
82	4129052	2015-12	735933
113	3748529	2016-01	735964
85	3543639	2016-02	735995
31	4137679	2016-03	736024

129 rows × 3 columns

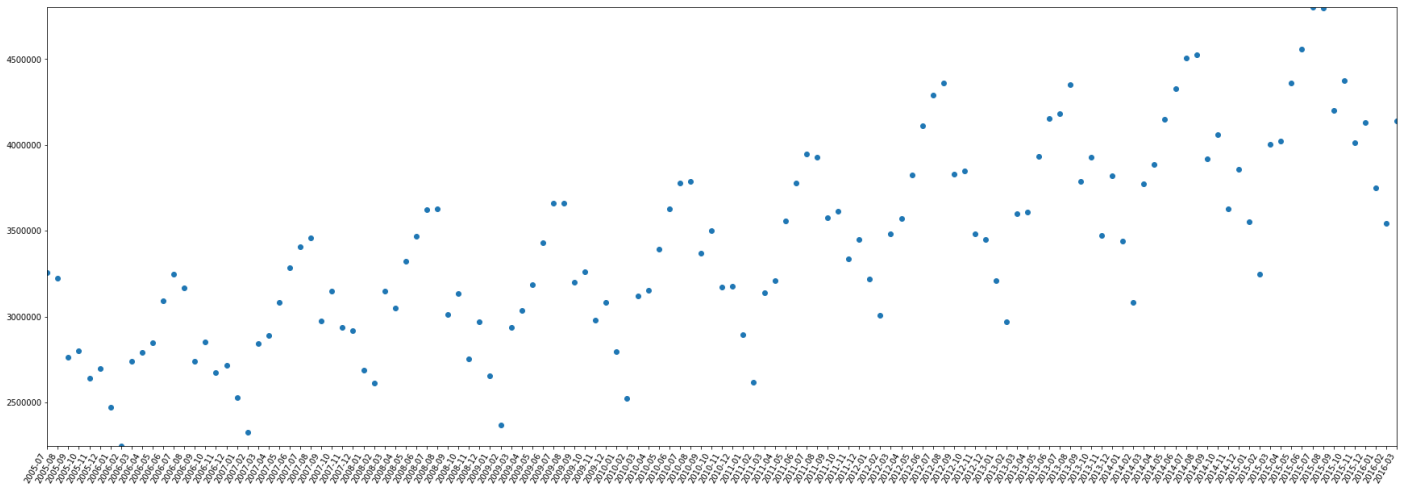
Observamos la correlación entre las variables de nuestro nuevo DataFrame:

```
[117] df_pass_by_year_pd_simple.corr().style.background_gradient(cmap='coolwarm')
```

	PassengerCountSum	DateOrd
PassengerCountSum	1.000000	0.773315
DateOrd	0.773315	1.000000

Vemos que existe una correlación positiva fuerte entre el número de pasajeros y el valor ordinal que representa nuestra fecha. Visualizamos esto con un gráfico de dispersión:

```
plt.scatter(df_pass_by_year_pd_simple["Date"], df_pass_by_year_pd_simple["AdjustedPassengerCountSum"])
plt.rcParams["figure.figsize"] = (30,10)
plt.xticks(rotation=60,ha="right")
plt.ticklabel_format(style='sci', axis='y', scilimits=(-1000000,1000000))
plt.margins(0)
plt.show()
```



Creamos nuestro modelo de regresión lineal utilizando el módulo sklearn:

```
model = linear_model.LinearRegression()
```

Creamos dos variables. Una para nuestro eje X y otra para nuestro eje Y:

```
explicativas = df_pass_by_year_pd_simple[['DateOrd']] #independiente
objetivo = df_pass_by_year_pd_simple[['PassengerCountSum']] #dependiente
```

Ambas variables son arrays de 2 dimensiones que utilizaremos para crear la regresión

Explicativas contiene los ordinales correspondientes a nuestras fechas, es decir, nuestra variable independiente.

Objetivo contiene aquello que nosotros queremos predecir, el número de pasajeros.

Utilizando el modelo creado anteriormente, creamos la regresión con la función **fit()**:

```
model.fit(explicativas , objetivo)
```

Haciendo uso de `__dict__` comprobamos los atributos de nuestro modelo:

```
model.__dict__  
  
{'fit_intercept': True,  
 'normalize': 'deprecated',  
 'copy_X': True,  
 'n_jobs': None,  
 'positive': False,  
 'n_features_in_': 1,  
 'coef_': array([[385.52307302]]),  
 '_residues': array([1.65561339e+13]),  
 'rank_': 1,  
 'singular_': array([12873.05356763]),  
 'intercept_': array([-2.79601791e+08]),  
 'feature_names_in_': array(['DateOrd'], dtype=object)}
```

Pasamos a realizar la predicción, para lo cual utilizamos el método **predict()** pasándole el valor de nuestras X, es decir nuestras fechas:

```
pred = model.predict(X=df_pass_by_year_pd_simple[['DateOrd']])
```

Esto nos devuelve un array con los valores de Y, es decir, nuestro número de pasajeros:

```
pred[:10]  
  
array([[2650445.11910808],  
       [2662396.33437181],  
       [2674347.54963553],  
       [2685913.24182624],  
       [2697864.45708996],  
       [2709430.14928061],  
       [2721381.36454433],  
       [2733332.57980806],  
       [2744127.22585267],  
       [2756078.44111639]])
```

Añadimos estos datos a nuestro DataFrame y lo ordenamos en función del campo DateOrd:

```
df_pass_by_year_pd_simple.insert(3, 'Prediction' , pred)

pd.set_option('display.float_format', '{:.3f}'.format)
df_pass_by_year_pd_simple = df_pass_by_year_pd_simple.sort_val-
ues(by=["DateOrd"])
df_pass_by_year_pd_simple
```

```
▶ pd.set_option('display.float_format', '{:.3f}'.format)
df_pass_by_year_pd_simple = df_pass_by_year_pd_simple.sort_val-
ues(by=["DateOrd"])
df_pass_by_year_pd_simple
```

	PassengerCountSum	Date	DateOrd	Prediction
60	3225769	2005-07-01	732128	2650445.119
62	3195866	2005-08-01	732159	2662396.334
90	2740553	2005-09-01	732190	2674347.550
117	2770715	2005-10-01	732220	2685913.242
84	2617333	2005-11-01	732251	2697864.457
...
109	4013814	2015-11-01	735903	4105794.720
82	4129052	2015-12-01	735933	4117360.412
113	3748529	2016-01-01	735964	4129311.627
85	3543639	2016-02-01	735995	4141262.842
31	4137679	2016-03-01	736024	4152443.012

129 rows x 4 columns

Comprobaremos la precisión nuestro modelo con la función **score()**:

```
[94] print(model.score(X=explicativas , y=objetivo))

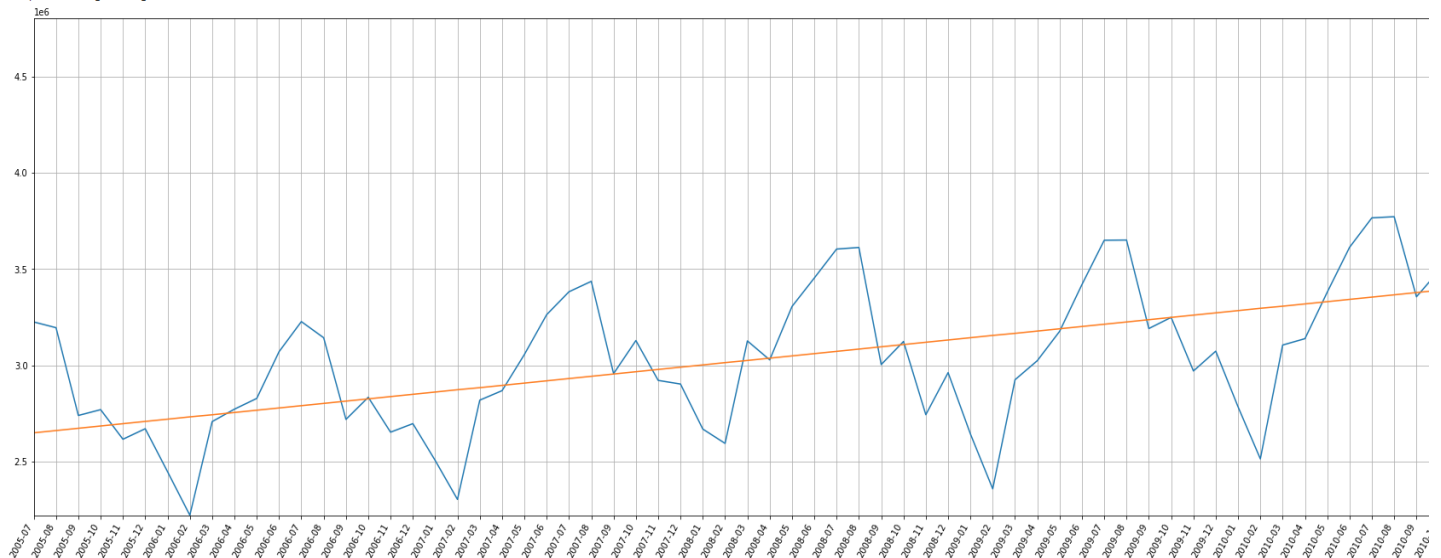
0.5980165190500482
```

Pasamos nuestras fechas de nuevo a formato AAAA-MM y visualizamos nuestra recta de regresión:

```
df_pass_by_year_pd_simple['Date'] =  
pd.to_datetime(df_pass_by_year_pd['Date']).dt.strftime("%Y-%m") #esto  
hace la columna un string
```

```
plt.plot(df_pass_by_year_pd_simple['Date'], df_pass_by_year_pd_simple['PassengerCountSum'], label="Datos reales")  
plt.plot(df_pass_by_year_pd_simple['Date'], df_pass_by_year_pd_simple['Prediction'], label="Predicción")  
plt.rcParams["figure.figsize"] = (60,10)  
plt.grid(True)  
plt.xticks(rotation=60,ha="right")  
plt.margins(0)  
plt.legend()
```

<matplotlib.legend.Legend at 0x7f9a8d6ca400>



Por último, generamos una lista de fechas a futuro para comprobar las predicciones de nuestro modelo. Estas fechas serán las correspondientes a todo el año 2016:

```
dates_list = pd.date_range('2016-01-01', '2016-12-31',  
                           freq='MS')  
df_pred_future = pd.DataFrame(dates_list, columns=["Date"])  
  
df_pred_future['DateOrd'] = df_pred_future['Date'].map(dt.datetime.toordinal)  
  
df_pred_future = df_pred_future.sort_values(by=["DateOrd"])
```

```
df_pred_future = pd.DataFrame(dates_list, columns=["Date"])  
  
df_pred_future['DateOrd'] = df_pred_future['Date'].map(dt.datetime.toordinal)  
df_pred_future = df_pred_future.sort_values(by=["DateOrd"])  
df_pred_future
```

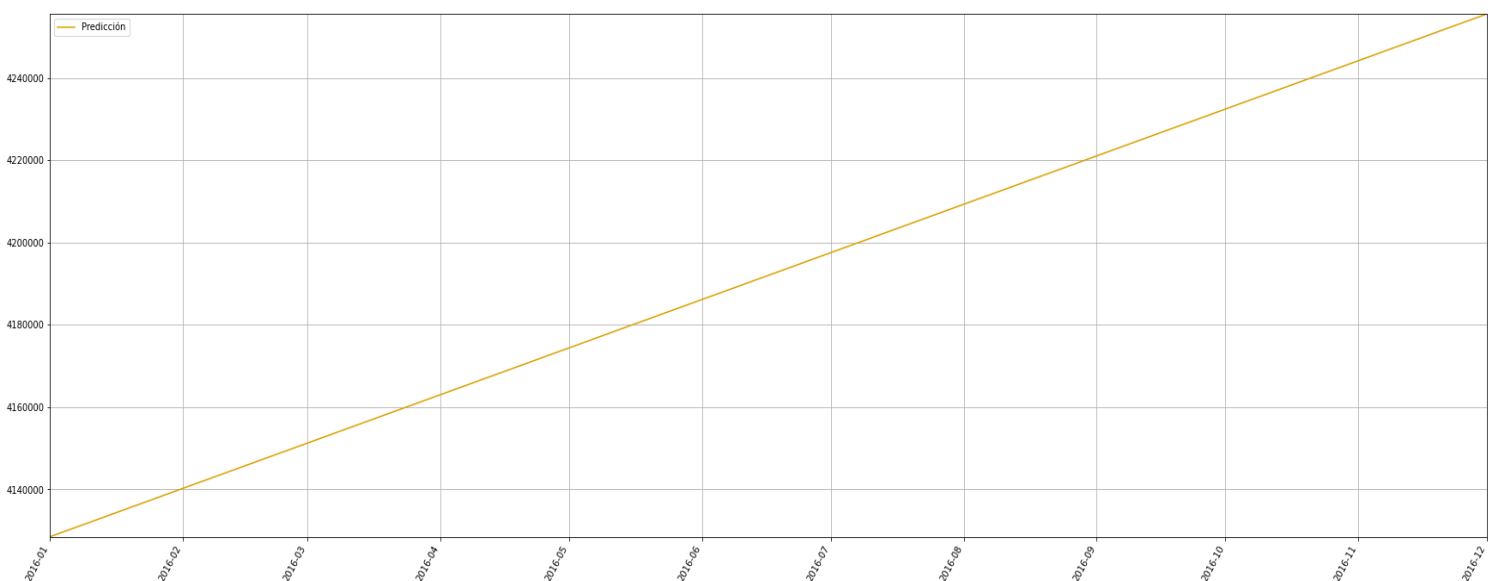
	Date	DateOrd
0	2016-01-01	735964
1	2016-02-01	735995
2	2016-03-01	736024
3	2016-04-01	736055
4	2016-05-01	736085
5	2016-06-01	736116
6	2016-07-01	736146
7	2016-08-01	736177
8	2016-09-01	736208
9	2016-10-01	736238
10	2016-11-01	736269
11	2016-12-01	736299

Con estas fechas generamos la predicción con el modelo ya creado:

```
pred_future = model.predict(X=df_pred_future[['DateOrd']])  
df_pred_future.insert(2, 'Prediction', pred_future)
```

```
[232] df_pred_future
```

	Date	DateOrd	Prediction
0	2016-01-01	735964	4128476.487
1	2016-02-01	735995	4140239.358
2	2016-03-01	736024	4151243.335
3	2016-04-01	736055	4163006.206
4	2016-05-01	736085	4174389.630
5	2016-06-01	736116	4186152.501
6	2016-07-01	736146	4197535.925
7	2016-08-01	736177	4209298.797
8	2016-09-01	736208	4221061.668
9	2016-10-01	736238	4232445.092
10	2016-11-01	736269	4244207.964
11	2016-12-01	736299	4255591.388



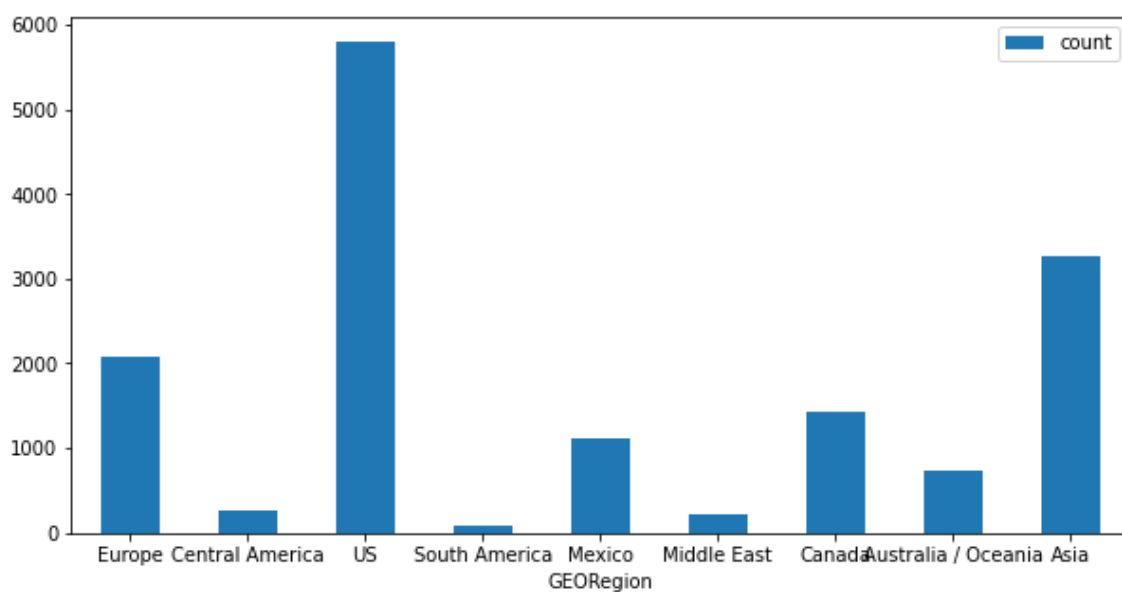


Observamos como la predicción nos indica que el número de pasajeros crecerá de manera lineal durante 2016.

CONCLUSIONES

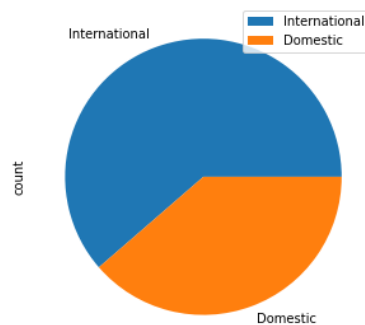
Realizando un análisis descriptivo de los datos hemos visto cómo podemos obtener una visión global de la información que estamos analizando y como, aunque parezca un análisis simple, podemos extraer conclusiones muy valiosas

Se puede observar, por ejemplo, que, de todas las regiones, Estados Unidos es la que ha recibido mayor número de vuelos:



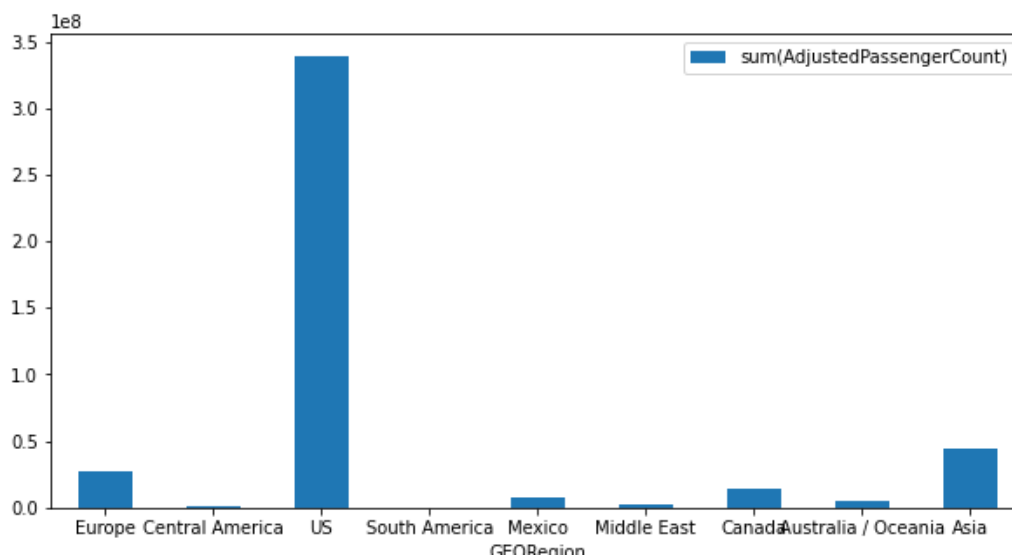
Sin embargo, vemos que la mayoría de vuelos son internacionales, es decir, hacia fuera de Estados Unidos.

```
+-----+
|  GEOSummary | count |
+-----+
| International | 9210 |
| Domestic     | 5797 |
+-----+
```



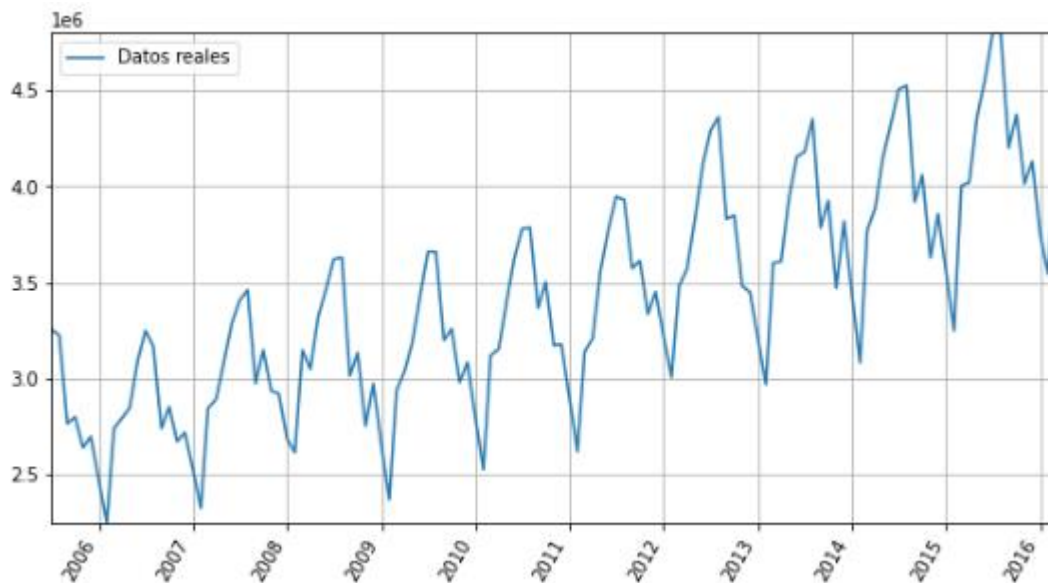
Aun con esto, Estados Unidos es la región con el mayor número de pasajeros, lo que quiere decir, que, aunque la mayoría de vuelos hayan sido internacionales (a otras regiones fuera de Estado Unidos), los vuelos domésticos transportaban a más pasajeros en total.

GEORegion	sum(AdjustedPassengerCount)
Europe	26695446
Central America	1355400
US	339042637
South America	250741
Mexico	8084752
Middle East	1852943
Canada	13901776
Australia / Oceania	4786892
Asia	44213493

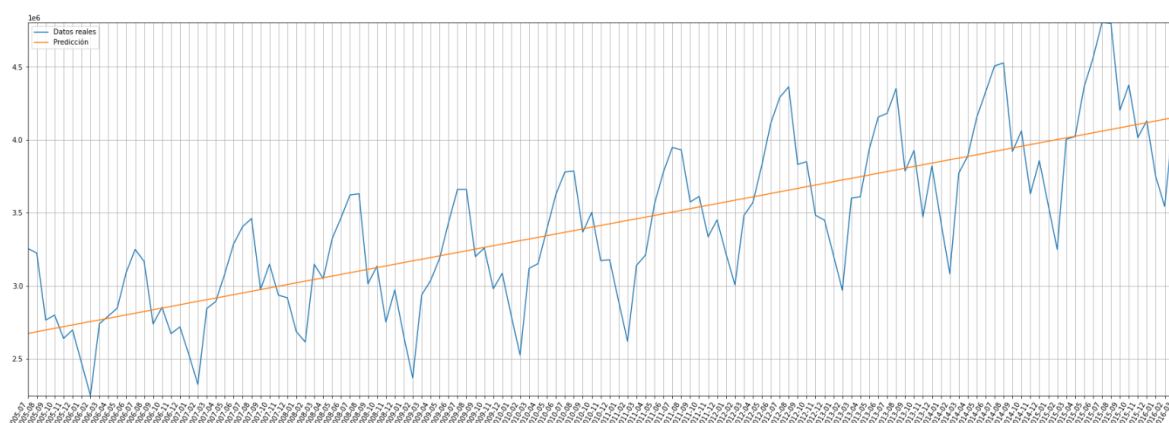


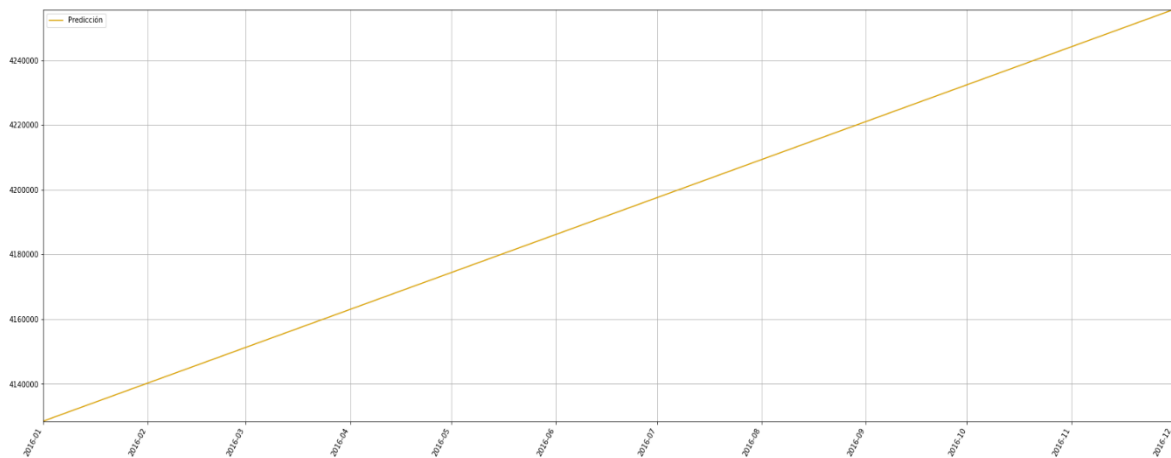
Con esto en cuenta (junto con otros factores) podríamos tomar decisiones, como destinar más recursos a realizar más vuelos domésticos, ya que hemos visto que son los que más pasajeros mueven o, podemos intentar destinar recursos a aumentar el interés de nuestros pasajeros por viajar al extranjero con tarifas más atractivas.

También, estudiado el número total de pasajeros de nuestro aeropuerto hemos observado como este número sigue un patrón, alcanzando su máximo y su mínimo en Agosto y Febrero de cada año respectivamente.



Para predecir como va a evolucionar este número se pueden utilizar diferentes algoritmos. En este estudio se ha realizado una regresión lineal y si solo nos fijásemos en ella, veríamos que este número va a ir en aumento de manera lineal.





Sin embargo, únicamente mirando la regresión lineal no podemos saber en qué momento el número de pasajeros va a descender o ascender bruscamente. Esto, aunque es imposible saberlo con total certeza, lo podemos suponer mirando los datos de años anteriores. De este modo, aunque la regresión lineal no muestre un descenso del número de pasajeros, nosotros podemos suponer, que, en este año, dicho número va a tener una tendencia ascendente hasta Julio/Agosto donde alcanzará un pico y a partir de ahí comenzará descender hasta el mes de Febrero como viene ocurriendo en los años anteriores. En función de esto podemos tomar decisiones.

Como conclusión podemos decir que un análisis descriptivo o una regresión lineal, por sí solos, no nos sirven para tomar decisiones o predecir comportamientos, es necesario analizar bien la información con varias técnicas en conjunto para tomar las mejores decisiones posibles basadas en los datos.



ANEXOS

1. Instalación DataStax Bulk Loader.

DataStax Bulk Loader (DSBulk) es una utilidad que permite la carga, descarga y conteo de datos de nuestra base de datos de manera rápida y eficaz desde la consola de comandos. Para su instalación se seguirán los siguientes pasos:

Desde la consola de comandos descargamos el archivo de instalación de DSBulk:

```
curl -OL https://downloads.datastax.com/dsbulk/dsbulk-1.9.0.tar.gz
```

```
C:\dsbulk-1.9.0\bin>curl -OL https://downloads.datastax.com/dsbulk/dsbulk-1.9.0.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100 31.5M  100 31.5M    0     0 3595k      0  0:00:08  0:00:08 --:--:-- 4618k
```

Una vez descargado el archivo de instalación lo descomprimos:

```
tar -xzvf dsbulk-1.9.0.tar.gz
```

Accedemos a la carpeta bin y desde la consola de comandos comprobamos que todo se ha instalado correctamente comprobando la versión de DSBulk

```
dsbulk-1.9.0/bin/>dsbulk --version
```

```
C:\dsbulk-1.9.0\bin>dsbulk --version
DataStax Bulk Loader v1.9.0
```


2. Extracción de datos. Vuelos de la compañía Air China.

Se muestra a continuación los datos de los vuelos de la compañía Air China. Se adjunta también a la entrega la carpeta *airchina_data_extraction* con el resultado de la extracción:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
activity_period	operating_airline	operating_airline_iata_code	published_airline	published_airline_iata_code	geo_summary	geo_region	activity_type_code	price_category_code	terminal	boarding_area	passenger_count	adjusted_activity_type_code	year	month
1	201203 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6957	Enplaned	2012	March
2	200905 Air China	CA	Air China	CA	International	Asia	Deplaned	Other	International	G	4880	Deplaned	2009	May
4	201004 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7290	Enplaned	2010	April
5	200711 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6991	Enplaned	2007	November
6	201509 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9430	Enplaned	2015	September
7	201306 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7678	Enplaned	2013	June
8	200702 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4426	Enplaned	2007	February
9	201002 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4779	Enplaned	2010	February
10	200712 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6769	Enplaned	2007	December
11	201505 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8801	Enplaned	2015	May
12	201206 Air China	CA	Air China	CA	International	Asia	Deplaned	Other	International	G	7211	Deplaned	2012	June
13	201310 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7718	Enplaned	2013	October
14	200509 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4568	Enplaned	2005	September
15	200701 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4845	Enplaned	2007	January
16	201009 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6905	Enplaned	2010	September
17	201301 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7941	Enplaned	2013	January
18	200901 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5560	Enplaned	2009	January
19	201011 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6586	Enplaned	2010	November
20	201508 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	10194	Enplaned	2015	August
21	200705 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7592	Enplaned	2007	May
22	201107 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7627	Enplaned	2011	July
23	201001 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5614	Enplaned	2010	January
24	201304 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8737	Enplaned	2013	April
25	200603 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4514	Enplaned	2006	March
26	200812 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4853	Enplaned	2008	December
27	200807 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6789	Enplaned	2008	July
28	201010 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7051	Enplaned	2010	October
29	201205 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7713	Enplaned	2012	May
30	201109 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7151	Enplaned	2011	September
31	201601 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9819	Enplaned	2016	January
32	200805 Air China	CA	Air China	CA	International	Asia	Deplaned	Other	International	G	6592	Deplaned	2008	May
33	201506 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9438	Enplaned	2015	June
34	200811 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5596	Enplaned	2008	November
35	201102 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6053	Enplaned	2011	February
36	200909 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5658	Enplaned	2009	September
37	201108 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7432	Enplaned	2011	August
38	200911 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4849	Enplaned	2009	November
39	201207 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7645	Enplaned	2012	July
40	200607 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5861	Enplaned	2006	July
41	201303 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7689	Enplaned	2013	March
42	200910 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5477	Enplaned	2009	October
43	201007 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7392	Enplaned	2010	July
44	200601 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	3470	Enplaned	2006	January
45	201504 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7465	Enplaned	2015	April
46	201307 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8771	Enplaned	2013	July
47	201308 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9346	Enplaned	2013	August
48	201406 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8284	Enplaned	2014	June
49	201005 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7608	Enplaned	2010	May
50	200808 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6408	Enplaned	2008	August
51	200709 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7209	Enplaned	2007	September
52	201003 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5277	Enplaned	2010	March
53	201410 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7446	Enplaned	2014	October
54	201012 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7194	Enplaned	2010	December
55	201210 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7425	Enplaned	2012	October
56	200610 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6801	Enplaned	2006	October
57	200707 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7874	Enplaned	2007	July
58	201603 Air China	CA	Air China	CA	International	Asia	Deplaned	Other	International	G	8021	Deplaned	2016	March
59	201408 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8661	Enplaned	2014	August
60	201211 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7085	Enplaned	2012	November
61	200706 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7273	Enplaned	2007	June
62	201101 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7271	Enplaned	2011	January
63	201201 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7482	Enplaned	2012	January
64	201305 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8620	Enplaned	2013	May
65	201412 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7713	Enplaned	2014	December
66	200507 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6263	Enplaned	2005	July
67	200512 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4261	Enplaned	2005	December
68	201112 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7484	Enplaned	2011	December
69	200511 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4550	Enplaned	2005	November
70	201106 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7111	Enplaned	2011	June
71	200508 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4204	Enplaned	2005	August
72	201403 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7971	Enplaned	2014	March
73	200907 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5796	Enplaned	2009	July
74	201511 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8925	Enplaned	2015	November
75	201309 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8581	Enplaned	2013	September
76	201501 Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7177	Enplaned	2015	January

77	200803	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7182	Enplaned	2008	March
78	201512	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9341	Enplaned	2015	December
79	200806	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6812	Enplaned	2008	June
80	200908	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5790	Enplaned	2009	August
81	200704	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6631	Enplaned	2007	April
82	200909	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5107	Enplaned	2009	September
83	200708	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8175	Enplaned	2007	August
84	200903	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4441	Enplaned	2009	March
85	200611	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5315	Enplaned	2006	November
86	200809	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5348	Enplaned	2008	September
87	200902	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	3972	Enplaned	2009	February
88	201006	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6992	Enplaned	2010	June
89	201105	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7497	Enplaned	2011	May
90	201111	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7188	Enplaned	2011	November
91	200908	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5826	Enplaned	2009	August
92	201404	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7791	Enplaned	2014	April
93	201204	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7078	Enplaned	2012	April
94	201110	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7145	Enplaned	2011	October
95	201312	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8179	Enplaned	2013	December
96	201209	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7356	Enplaned	2012	September
97	201510	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9208	Enplaned	2015	October
98	200710	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7678	Enplaned	2007	October
99	201104	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7014	Enplaned	2011	April
100	200602	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	2862	Enplaned	2006	February
101	201402	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7861	Enplaned	2014	February
102	201202	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6173	Enplaned	2012	February
103	201405	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8038	Enplaned	2014	May
104	201212	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7467	Enplaned	2012	December
105	201103	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7188	Enplaned	2011	March
106	201401	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	9318	Enplaned	2014	January
107	200612	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4805	Enplaned	2006	December
108	201503	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7836	Enplaned	2015	March
109	200801	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7603	Enplaned	2008	January
110	201208	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7975	Enplaned	2012	August
111	201507	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	10316	Enplaned	2015	July
112	200610	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5855	Enplaned	2006	October
113	201502	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7023	Enplaned	2015	February
114	200802	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5785	Enplaned	2008	February
115	200606	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5920	Enplaned	2006	June
116	201409	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	8187	Enplaned	2014	September
117	201407	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6916	Enplaned	2014	July
118	201311	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7814	Enplaned	2013	November
119	200510	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4715	Enplaned	2005	October
120	200701	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5902	Enplaned	2007	March
121	200906	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5348	Enplaned	2009	June
122	200604	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	4728	Enplaned	2006	April
123	200804	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6903	Enplaned	2008	April
124	200912	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5448	Enplaned	2009	December
125	201602	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7731	Enplaned	2016	February
126	200605	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5680	Enplaned	2006	May
127	201411	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6666	Enplaned	2014	November
128	200904	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	5533	Enplaned	2009	April
129	201008	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	7823	Enplaned	2010	August
130	201302	Air China	CA	Air China	CA	International	Asia	Enplaned	Other	International	G	6652	Enplaned	2013	February

3. Extracción de datos. Vuelos de la compañía Air Berlín.

Se muestra a continuación los datos de los vuelos de la compañía Air Berlin. Se adjunta también a la entrega la carpeta *airberlin_data_extraction* con el resultado de la extracción:

1	activity_period	operating_airline	operating_airline_jata_code	published_airline	published_airline_jata_code	geo_summary	geo_region	activity_type_code	price_category_code	terminal	boarding_area	passenger_count	adjusted_activity_type_code	year	month
2	201009	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	2357	Enplaned	2010	September
3	201008	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	2294	Enplaned	2010	August
4	201007	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	2620	Enplaned	2010	July
5	201010	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	1689	Enplaned	2010	October
6	201005	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	1337	Enplaned	2010	May
7	201006	Air Berlin	AB	Air Berlin	AB	International	Europe	Enplaned	Other	International	G	2548	Enplaned	2010	June

4. Extracción de datos. Número de compañías diferentes.

Se muestra a continuación las diferentes compañías existentes. Se adjunta también a la entrega la carpeta *airport_companies* con el resultado de la extracción:



1	OperatingAirline
2	Icelandair
3	Ameriflight
4	Cathay Pacific
5	Aeromexico
6	Etihad Airways
7	Philippine Airlines
8	United Airlines - Pre 07/01/2013
9	Turkish Airlines
10	Swiss International
11	Independence Air
12	Miami Air International
13	Air France
14	Japan Airlines
15	Midwest Airlines
16	Atlas Air, Inc
17	JetBlue Airways
18	China Eastern
19	Mexicana Airlines
20	Air Canada
21	Allegiant Air
22	Northwest Airlines
23	Sun Country Airlines
24	Evergreen International Airlines
25	Horizon Air
26	United Airlines
27	World Airways
28	Virgin America
29	Jet Airways
30	British Airways
31	Swissport USA
32	Servisair
33	Hawaiian Airlines
34	Virgin Atlantic
35	SAS Airlines
36	TACA
37	KLM Royal Dutch Airlines
38	Boeing Company



39	ExpressJet Airlines
40	Compass Airlines
41	Lufthansa German Airlines
42	SkyWest Airlines
43	Asiana Airlines
44	China Airlines
45	Frontier Airlines
46	American Airlines
47	Air China
48	Air Berlin
49	Delta Air Lines
50	Korean Air Lines
51	AirTran Airways
52	All Nippon Airways
53	Singapore Airlines
54	China Southern
55	US Airways
56	Air Canada Jazz
57	Emirates
58	Pacific Aviation
59	Air India Limited
60	Atlantic Southeast Airlines
61	American Eagle Airlines
62	Qantas Airways
63	COPA Airlines, Inc.
64	Alaska Airlines
65	Xtra Airways
66	Republic Airlines
67	Mesaba Airlines
68	Air New Zealand
69	Spirit Airlines
70	ATA Airlines
71	BelAir Airlines
72	Aer Lingus
73	WestJet Airlines
74	Mesa Airlines
75	LAN Peru
76	Southwest Airlines
77	XL Airways France
78	EVA Airways



5. Extracción de datos. Media de pasajeros por compañía.

Se muestra a continuación la media de pasajeros por compañía. Se adjunta también a la entrega la carpeta *avg_flights_per_company* con el resultado de la extracción:



1	OperatingAirline	avg(PassengerCount)	avg(AdjustedPassengerCount)
2	Icelandair	2799,70000	2799,70000
3	Ameriflight	5,00000	5,36364
4	Cathay Pacific	17121,32558	17121,32558
5	Aeromexico	5463,82222	5463,82222
6	Etihad Airways	6476,08824	6476,08824
7	Philippine Airlines	10248,63566	10248,63566
8	United Airlines - Pre 07/01/2013	48915,46750	49365,51671
9	Turkish Airlines	8162,41667	8162,41667
10	Swiss International	6061,64029	6061,64029
11	Independence Air	6391,30000	6391,30000
12	Miami Air International	107,37500	107,37500
13	Air France	11589,07752	11589,07752
14	Japan Airlines	6470,33205	6471,42857
15	Midwest Airlines	3883,00000	3883,00000
16	Atlas Air, Inc	34,00000	35,50000
17	JetBlue Airways	35261,13964	35261,13964
18	China Eastern	5498,40278	5498,40278
19	Mexicana Airlines	7993,80645	7993,80645
20	Air Canada	18251,56011	18251,56011
21	Allegiant Air	1516,81250	1516,81250
22	Northwest Airlines	26109,25000	26205,50417
23	Sun Country Airlines	3992,65200	3992,65200
24	Evergreen International Airlines	2,00000	2,00000
25	Horizon Air	5577,58333	5577,58333
26	United Airlines	72732,05830	72827,21973
27	World Airways	261,66667	261,66667
28	Virgin America	74405,35359	74405,35359
29	Jet Airways	4280,31250	4280,31250
30	British Airways	17625,12403	17625,12403
31	Swissport USA	258,60000	264,80000
32	Servisair	90,05556	90,05556
33	Hawaiian Airlines	8282,18605	8282,18605
34	Virgin Atlantic	9847,10465	9847,10465
35	SAS Airlines	5865,84722	5865,84722
36	TACA	5066,19767	5066,19767
37	KLM Royal Dutch Airlines	9221,81395	9221,81395
38	Boeing Company	18,00000	18,00000



39	ExpressJet Airlines	5631,84375	5631,84375
40	Compass Airlines	23358,55682	23359,84091
41	Lufthansa German Airlines	19301,96512	19301,96512
42	SkyWest Airlines	37083,83904	37083,87643
43	Asiana Airlines	5902,96124	5902,96124
44	China Airlines	9857,51550	9857,51550
45	Frontier Airlines	17787,67692	17787,67692
46	American Airlines	127164,38971	127164,38971
47	Air China	6618,33591	6618,33591
48	Air Berlin	2320,75000	2320,75000
49	Delta Air Lines	68498,49741	68515,41969
50	Korean Air Lines	5678,46124	5678,46124
51	AirTran Airways	10569,23894	10569,23894
52	All Nippon Airways	6385,52326	6385,52326
53	Singapore Airlines	14746,64729	14746,64729
54	China Southern	4321,43750	4321,43750
55	US Airways	55317,81579	55317,81579
56	Air Canada Jazz	294,21429	294,21429
57	Emirates	9070,86667	9070,86667
58	Pacific Aviation	160,00000	160,00000
59	Air India Limited	2834,50000	2834,50000
60	Atlantic Southeast Airlines	2176,90909	2176,90909
61	American Eagle Airlines	4006,52830	4006,52830
62	Qantas Airways	4991,21642	4991,21642
63	COPA Airlines, Inc.	3418,07143	3418,07143
64	Alaska Airlines	17251,63782	17564,67776
65	Xtra Airways	73,00000	73,00000
66	Republic Airlines	2452,50000	2452,50000
67	Mesaba Airlines	2864,72727	2864,72727
68	Air New Zealand	7452,33977	7452,33977
69	Spirit Airlines	2921,04167	2921,04167
70	ATA Airlines	8744,63636	9661,65909
71	BelAir Airlines	415,36364	428,00000
72	Aer Lingus	4407,18367	4407,18367
73	WestJet Airlines	5338,15534	5338,15534
74	Mesa Airlines	3710,58120	3710,58120
75	LAN Peru	2786,01111	2786,01111
76	Southwest Airlines	81188,15858	81223,34951
77	XL Airways France	2223,16129	2240,12903
78	EVA Airways	13116,35659	13116,35659

6. Extracción de datos. Eliminación de duplicados por GEORegion.

Se muestra a continuación los datos correspondientes a eliminar las duplicidades por el campo GEORegion manteniendo solo aquellos registros con mayor número de pasajeros. Se adjunta también a la entrega la carpeta *airtraffic_drop_duplicates_georegion* con el resultado de la extracción:

1	ActivityPeriod	OperatingAirline	OperatingAirlineIATACode	PublishedAirline	PublishedAirlineIATACode	GEOSummary	GEORegion	ActivityTypeCode	PriceCategoryCode	Terminal	BoardingArea	PassengerCount	AdjustedActivityTypeCode	AdjustedPassengerCount	Year	Month
2	200706	Air Canada	AC	Air Canada	AC	International	Canada	Deplaned	Other	Terminal 3	E	39798	Deplaned	39798	2007	August
3	200706	United Airlines - Pre 07/01/2013	UA	United Airlines - Pre 07/01/2013	UA	International	Asia	Deplaned	Other	International	G	86598	Deplaned	86598	2007	August
4	201101	LAN Peru	LP	LAN Peru	LP	International	South America	Deplaned	Other	International	A	3685	Deplaned	3685	2011	January
5	201308	United Airlines	UA	United Airlines	UA	Domestic	US	Deplaned	Other	Terminal 3	F	659837	Deplaned	659837	2013	August
6	201407	United Airlines	UA	United Airlines	UA	International	Mexico	Deplaned	Other	International	G	29206	Deplaned	29206	2014	July
7	201410	TACA	TA	TACA	TA	International	Central America	Deplaned	Other	International	A	8970	Deplaned	8970	2014	October
8	201501	Air New Zealand	NZ	Air New Zealand	NZ	International	Australia / Oceania	Enplaned	Other	International	G	12973	Enplaned	12973	2015	January
9	201507	Emirates	EX	Emirates	EX	International	Middle East	Deplaned	Other	International	A	14769	Deplaned	14769	2015	July
10	201507	United Airlines	UA	United Airlines	UA	International	Europe	Deplaned	Other	International	G	48136	Deplaned	48136	2015	July