

DOCUMENTATION: EXPLORING INSIGHTS FROM SYNTHETIC AIRLINE DATA ANALYSIS WITH QLIK

Table of Contents

1. Project Flow(#project-flow)
2. Data Collection and Extraction(#data-collection-and-extraction)
3. Data Preparation(#data-preparation)
4. Data Visualization(#data-visualization)
5. Dashboard Development(#dashboard-development)
6. Story Creation(#story-creation)
7. Performance Testing(#performance-testing)
8. Project Demonstration(#project-demonstration)
9. Documentation and Reporting(#documentation-and-reporting)

Project Flow:

Overview:

The project is structured to analyze synthetic airline data using Qlik's robust data analytics and visualization capabilities. The project flow consists of the following phases:

1. Data Collection and Extraction: Aggregating synthetic data from various sources.
2. Data Preparation: Cleaning and transforming data for analysis.
3. Data Visualization: Creating visual representations of data insights.
4. Dashboard Development: Building interactive dashboards in Qlik.
5. Story Creation: Crafting narrative stories to present findings.
6. Performance Testing: Ensuring the efficiency and responsiveness of Qlik applications.
7. Project Demonstration: Showcasing the final product.
8. Documentation and Reporting: Compiling comprehensive project reports and user guides.

Objectives

- Derive insights into revenue optimization, operational efficiency, and customer experience enhancement.
- Utilize Qlik to create actionable data visualizations and dashboards.

Data Collection and Extraction

Data Sources

The synthetic airline data comprises:

- Flight Schedules: Information on flight timings, routes, and frequencies.
- Passenger Demographics: Age, gender, and other relevant attributes of passengers.
- Ticket Sales: Historical sales data including pricing and booking patterns.
- Performance Metrics: On-time performance, delays, and cancellations.

Extraction Process

1. Database Access: Establish connections to the synthetic data repositories.
2. Querying: Use SQL or other query languages to extract relevant data subsets.
3. Data Export: Export data in formats compatible with Qlik (e.g., CSV, Excel, JSON).

Tools

- SQL Clients: For querying and extracting data from relational databases.
- Data Export Tools: Built-in or third-party tools for exporting data.

Data Preparation

Data Cleaning

1. Handling Missing Values: Impute or remove incomplete records.
2. Data Normalization: Standardize formats (dates, numeric values).
3. Data Validation: Ensure accuracy and consistency of data fields.

Data Transformation

1. Aggregating Data: Summarize data to the required granularity (daily, monthly).
2. Feature Engineering: Create new variables or metrics for deeper analysis.

Tools

- ETL Tools: Tools like Qlik Sense, Talend, or custom scripts for data transformation.
- Data Profiling Tools: For assessing data quality and structure.

Data Visualization

Visualization Goals

- Identify Trends: Recognize patterns in ticket sales, flight schedules, etc.
- Spot Correlations: Understand relationships between variables (e.g., pricing and demand).

Visualization Types

1. Time Series Charts: For visualizing trends over time (e.g., revenue trends).
2. Heatmaps: To display flight frequency and passenger flow data.
3. Bar and Pie Chart: For categorical data such as destination popularity.
4. Scatter Plots: To analyze correlations between multiple variables.

Tools

- Qlik Sense: For creating interactive visualizations and dashboards.

Dashboard Development

Dashboard Design

1. User Interface: Design intuitive and accessible dashboards.
2. Interactivity: Include filters and drill-down capabilities for dynamic data exploration.
3. KPIs: Highlight key performance indicators for quick insights.

Implementation Steps

1. Data Integration: Load cleaned and transformed data into Qlik.
2. Visualization Creation: Develop individual charts and graphs.
3. Dashboard Assembly: Combine visualizations into coherent dashboards.

Tools

- Qlik Sense: Central tool for dashboard creation and data visualization.

DASHBOARD ANALYSIS:

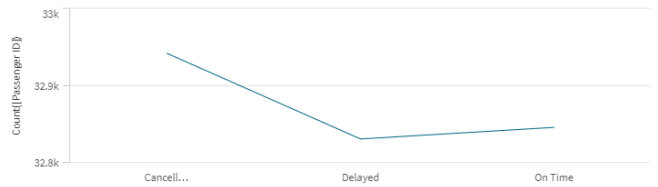
AIRLINE ANALYSIS

PASSENGER COUNT

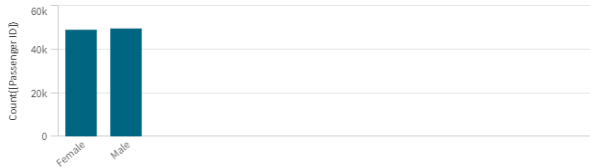
Count([Passenger ID])

98.62k

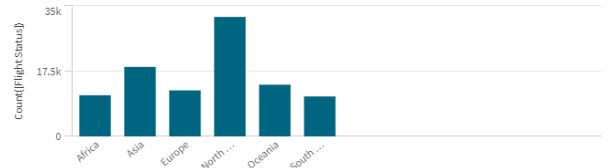
FLIGHT STATUS



MALE AND FEMALE PASSENGERS

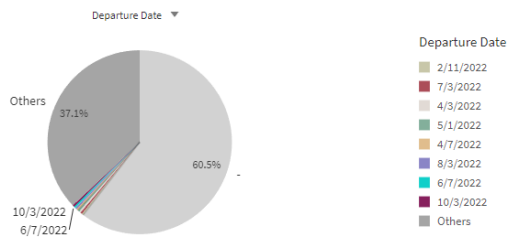


CONTINENT WISE FLIGHT STATUS

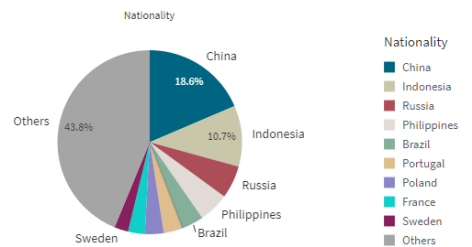


AIRLINE DOING

OPERATIONAL EFFICIENCY

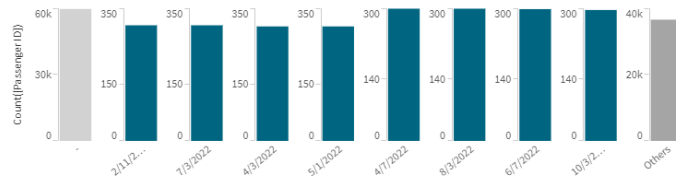


NATIONALITY WISE PASSENGER

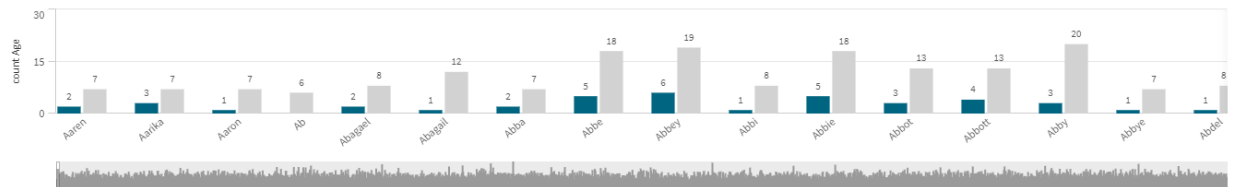


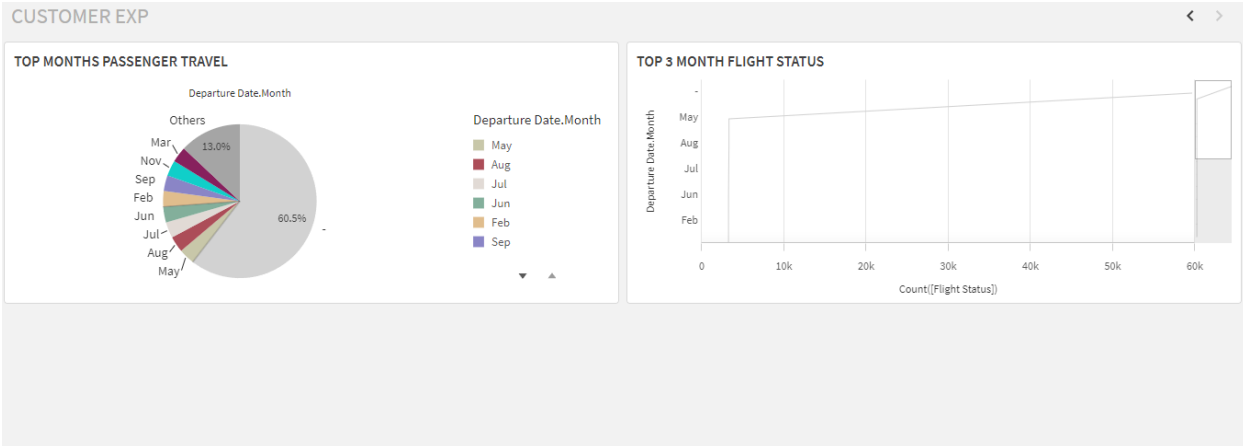
AIRLINE DATA ANALYSIS

REVENUE OPTIMIZATION



count Age by Departure Date (YTD) and First Name





Story Creation

Objectives

- Narrate Insights: Create compelling narratives to explain data findings.
- Support Decision-Making: Provide actionable insights and recommendations.

Components

1. Slides: Organize insights into a sequence of slides or pages.
2. Annotations: Include descriptive text and comments to explain visualizations.
3. Scenarios: Present different scenarios (e.g., revenue optimization strategies).

Tools

- Qlik Sense: Use built-in story features to create presentations.

Performance Testing

Goals

- Ensure Responsiveness: Dashboards and visualizations should load quickly.
- Test Scalability: Handle large data volumes efficiently.

Testing Methods

1. Load Testing: Simulate high user traffic and large data loads.
2. Stress Testing: Determine the breaking point of the system.
3. Performance Monitorin:: Use Qlik's performance tools to monitor response times and resource usage.

Tools

- Qlik Performance Analyzer: For evaluating the performance of Qlik applications.

Project Demonstration

Preparation

1. Demo Script: Prepare a script outlining the demonstration flow.
2. Use Cases: Select specific scenarios to demonstrate (e.g., peak travel time analysis).

Execution

1. Live Demo: Showcase dashboards and interactive features in a live setting.
2. Q&A Session: Address questions and feedback from the audience.

Tools

- Qlik Sense: Primary tool for the live demonstration.

Code used/ Commands:

```
SET ThousandSep='';  
SET DecimalSep='.';  
SET MoneyThousandSep='';  
SET MoneyDecimalSep='.';  
SET MoneyFormat='$ ###0.00;- $ ###0.00';  
SET TimeFormat='h:mm:ss TT';  
SET DateFormat='M/D/YYYY';  
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';  
SET FirstWeekDay=6;  
SET BrokenWeeks=1;  
SET ReferenceDay=0;  
SET FirstMonthOfYear=1;  
SET CollationLocale='en-US';  
SET CreateSearchIndexOnReload=1;  
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
```

SET

LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';

SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';

SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y';

Set dataManagerTables = "','Airline Dataset Updated - v2';

//This block renames script tables from non generated section which conflict with the names of managed tables

For each name in \$(dataManagerTables)

Let index = 0;

Let currentName = name;

Let tableNumber = TableNumber(name);

Let matches = 0;

Do while not IsNull(tableNumber) or (index > 0 and matches > 0)

index = index + 1;

currentName = name & '-' & index;

tableNumber = TableNumber(currentName)

matches = Match('\$(currentName)', \$(dataManagerTables));

Loop

If index > 0 then

Rename Table '\$(name)' to '\$(currentName)';

EndIf;

Next;

Set dataManagerTables = ;

Unqualify *;

__countryAliasesBase:

LOAD

Alias AS [__Country],

ISO3Code AS [__ISO3Code]

FROM [lib://DataFiles/countryAliases.qvd]

(qvd);

__countryGeoBase:

LOAD

ISO3Code AS [__ISO3Code],

```
        ISO2Code AS [__ISO2Code],
        Polygon AS [__Polygon]
FROM [lib://DataFiles/countryGeo.qvd]
(qvd);
```

```
__countryName2IsoThree:
MAPPING LOAD
    __Country,
    __ISO3Code
RESIDENT __countryAliasesBase;
```

```
__countryCodeIsoThree2Polygon:
MAPPING LOAD
    __ISO3Code,
    __Polygon
RESIDENT __countryGeoBase;
```

```
__countryCodeIsoTwo2Polygon:
MAPPING LOAD
    __ISO2Code,
    __Polygon
RESIDENT __countryGeoBase;
```

```
[Airline Dataset Updated - v2]:
LOAD
    [Passenger ID],
    [First Name],
    [Last Name],
    [Gender],
    [Age],
    [Nationality],
    [Airport Name],
    [Airport Country Code],
    [Country Name],
    [Airport Continent],
    [Continents],
    Date(Date#([Departure Date], 'MM-DD-YYYY') ) AS [Departure Date],
    [Arrival Airport],
    [Pilot Name],
    [Flight Status],
    APPLYMAP( '__countryCodeIsoThree2Polygon', APPLYMAP( '__countryName2IsoThree',
```



```

LOWER([Nationality])), '-') AS [Airline Dataset Updated - v2.Nationality_GeoInfo],
    APPLYMAP( '__countryCodeIsoTwo2Polygon', UPPER([Airport Country Code]), '-') AS
[Airline Dataset Updated - v2.Airport Country Code_GeoInfo],
    APPLYMAP( '__countryCodeIsoThree2Polygon', APPLYMAP( '__countryName2IsoThree',
LOWER([Country Name])), '-') AS [Airline Dataset Updated - v2.Country Name_GeoInfo]
FROM [lib://DataFiles/Airline Dataset Updated - v2.csv]
(txt, utf8, embedded labels, delimiter is ',', msq);

```

```

TAG FIELD [Nationality] WITH '$geoname', '$relates_Airline Dataset Updated -
v2.Nationality_GeoInfo';
TAG FIELD [Airline Dataset Updated - v2.Nationality_GeoInfo] WITH '$geopolygon', '$hidden',
'$relates_Nationality';
TAG FIELD [Airport Country Code] WITH '$geoname', '$relates_Airline Dataset Updated -
v2.Airport Country Code_GeoInfo';
TAG FIELD [Airline Dataset Updated - v2.Airport Country Code_GeoInfo] WITH '$geopolygon',
'$hidden', '$relates_Airport Country Code';
TAG FIELD [Country Name] WITH '$geoname', '$relates_Airline Dataset Updated - v2.Country
Name_GeoInfo';
TAG FIELD [Airline Dataset Updated - v2.Country Name_GeoInfo] WITH '$geopolygon', '$hidden',
'$relates_Country Name';

```

```

DROP TABLES __countryAliasesBase, __countryGeoBase;
[autoCalendar]:
    DECLARE FIELD DEFINITION Tagged ('$date')
FIELDS
    Dual(Year($1), YearStart($1)) AS [Year] Tagged ('$axis', '$year'),
    Dual('Q'&Num(Ceil(Num(Month($1))/3)),Num(Ceil(NUM(Month($1))/3),00)) AS [Quarter]
Tagged ('$quarter', '$cyclic'),
    Dual(Year($1)&'-'&Q'&Num(Ceil(Num(Month($1))/3)),QuarterStart($1)) AS [YearQuarter] Tagged
('$yearquarter', '$qualified'),
    Dual('Q'&Num(Ceil(Num(Month($1))/3)),QuarterStart($1)) AS [_YearQuarter] Tagged
('$yearquarter', '$hidden', '$simplified'),
    Month($1) AS [Month] Tagged ('$month', '$cyclic'),
    Dual(Year($1)&'-'&Month($1), monthstart($1)) AS [YearMonth] Tagged ('$axis', '$yearmonth',
'$qualified'),
    Dual(Month($1), monthstart($1)) AS [_YearMonth] Tagged ('$axis', '$yearmonth', '$simplified',
'$hidden'),
    Dual('W'&Num(Week($1),00), Num(Week($1),00)) AS [Week] Tagged ('$weeknumber', '$cyclic'),
    Date(Floor($1)) AS [Date] Tagged ('$axis', '$date', '$qualified'),

```

```

Date(Floor($1), 'D') AS [_Date] Tagged ('$axis', '$date', '$hidden', '$simplified'),
If (DayNumberOfYear($1) <= DayNumberOfYear(Today()), 1, 0) AS [InYTD] ,
Year(Today())-Year($1) AS [YearsAgo] ,
If (DayNumberOfQuarter($1) <= DayNumberOfQuarter(Today()),1,0) AS [InQTD] ,
4*Year(Today())+Ceil(Month(Today())/3)-4*Year($1)-Ceil(Month($1)/3) AS [QuartersAgo] ,
Ceil(Month(Today())/3)-Ceil(Month($1)/3) AS [QuarterRelNo] ,
If(Day($1)<=Day(Today()),1,0) AS [InMTD] ,
12*Year(Today())+Month(Today())-12*Year($1)-Month($1) AS [MonthsAgo] ,
Month(Today())-Month($1) AS [MonthRelNo] ,
If(WeekDay($1)<=WeekDay(Today()),1,0) AS [InWTD] ,
(WeekStart(Today())-WeekStart($1))/7 AS [WeeksAgo] ,
Week(Today())-Week($1) AS [WeekRelNo] ;

```

DERIVE FIELDS FROM FIELDS [Departure Date] USING [autoCalendar] ;

Conclusion

This documentation provides a structured approach to analyzing synthetic airline data using Qlik. By following the outlined steps, users can effectively derive insights into revenue optimization, operational efficiency, and customer experience, supporting better decision-making for airlines and airport authorities.