TECHNICAL REPORT

Business Reporting Tools

Submitted by:

Andres Olivera

Kripa Fernandes

Dinu Wijayaweera

Submitted to:

Professor: Meire Matthijs

Msc Big Data Analytics for Business - IESEG 2019/2020

Introduction

A merged table was created with the relevant information we were to use for the analysis. This was to allow easy of functioning in two ways – aliases were not required to be referenced every time the code would be run and to allow import of one table into Tableau and avoid using their merge function.

```
PROC SQL;

CREATE TABLE rrrrrr AS

SELECT f.carrier, f.month, f.tailnum, f.dep_delay, f.arr_delay, f.flight, f.origin, f.dest, f.distance,
f.time_hour, f.air_time,  w.temp, w.dewp, w.humid, w.wind_dir, w.wind_speed, w.wind_gust,

w.precip, w.pressure, w.visib, p.type, p.manufacturer, p.model, p.engines, p.seats, p.engine,
a.name AS airline_name,

ar.name AS airport_name, ar.lat AS Latitude, ar.lon AS Longitude, ar.alt AS Airport_Altitude,
p.year as year_manu, f.hour,put(f.Month, 15.) || put(f.Day, 15.) as monthday, case when f.month
= 1 then 'Jan'

when f.month = 2 then 'Feb'

when f.month = 3 then 'Mar'

when f.month = 4 then 'Apr'

when f.month = 5 then 'May'

when f.month = 6 then 'Jun'

when f.month = 7 then 'Jul'

when f.month = 8 then 'Aug'

when f.month = 9 then 'Sep'

when f.month = 10 then 'Oct'
when f.month = 11 then 'Nov'

when f.month = 12 then 'Dec'


end as month_name
    FROM g_a.flights AS f, g_a.weather AS w, g_a.planes AS p, g_a.airlines AS a, g_a.airports AS ar
    WHERE f.time_hour = w.time_hour
        AND f.dep_delay > 0
        AND f.arr_delay > 0
    AND f.tailnum = p.tailnum
    AND f.carrier = a.carrier
    AND f.origin = ar.faa /*not matched to destination*/
;
QUIT;
RUN;
```

In our analysis, it was evident that departure delays exacerbated arrival delays. This is origin was mapped to the FAA rather than destination, in the merged table.

WEATHER DATA

The different parameters in weather were analysed by dividing the data into different levels and grouping them accordingly. This was done using the MAX, MIN and AVG functions. This was linked only to the departure delays as the weather data was only available for the origin.

```
PROC SQL;
SELECT origin, max(humid) AS Max_humid, min(humid) AS Min_humid, avg(humid) AS Avg_humid, avg(dep_delay) AS Dep_delay, avg(arr_delay) AS Arr_delay
FROM g_a.F_W_P_A_AR4
GROUP BY 1;
```

```
QUIT;
RUN;
```

This allowed the user to get a glimpse of whether there was a correlation. If a correlation was found, the data was ultimately put on Tableau and a linear regression line was added to see the strength of the correlation. The code for the humidity parameter is given below. A similar format was used for the other parameters.

```
PROC SQL;
CREATE TABLE g_a.humid_delay AS
    SELECT (CASE WHEN humid > 90 THEN 'Extremely humid (Greater than 80)'
            WHEN humid > 50 THEN 'AVERAGE HUMIDITY (Greater than 50, less than 80)'
            ELSE 'LOW HUMIDITY (Less than 50)' END) AS Humidity_levels,
            AVG(dep_delay) AS Average_Departure_Delay
    FROM g_a.F_W_P_A_AR4
    GROUP BY 1;
QUIT;
RUN;
```

It was observed that departure delays increased with an increase in precipitation, humidity and dew point (positive correlation) and decreased with an increase in pressure and visibility. The weather data remained similar for all airports as they are located quite close to each other but there were visible outliers in terms of delay for JFK.

For the tableau analysis, in order to avoid calculations on tableau, separate tables were made for analysis. Similar code was repeated for other parameters. The code followed the following format:

```
PROC SQL;
create table g_a.humid_and_airport as
SELECT w.humid as Humidity, A.name as Departure_Airport,
    sum(F.dep_delay) as Departure_Delay,
    sum(F.air_time) as Total_air_time,
    sum(F.dep_delay)/count(F.time_hour) as Average_Delay_per_Flight
```

```
FROM g_a.Flights F, g_a.Weather W, g_a.Airports A

WHERE W.year = F.year

AND W.month = F.month

AND W.day = F.day

AND F.origin = A.faa

GROUP BY 1, 2

ORDER BY 1 desc;

QUIT;

RUN;
```

PLANE DATA ANALYSIS

Several queries were created to initially analyze data of the plane in relation to the flights and airports. This enabled to find if there are relationships with certain columns and tables belonging to the NYC database. The below query was used to identify if there is a relationship with the number of engines, model and the delay.

By using this query, it was identified that there was no significant relationship with the number of engines but there was a relationship with the engine model and the arrival delay. Thereby this query was used to showcase the delay with the model in the tableau project.

```
/*Delays due to engine and model*/

PROC SQL;

CREATE TABLE Eng_Delays AS

    SELECT F.time_hour, P.model, P.tailnum AS FlightNo, avg(P.engines) AS Avg_Engines,
F.arr_delay

    FROM planes_d.planes AS P, planes_d.flights AS F

        WHERE P.tailnum=f.tailnum

    GROUP BY P.model

    ORDER BY 5 DESC;

 QUIT;

RUN;
```

The Speed related data was limited and initially we decided on not using. However, we decided to analyze the available data by an inner join. Here we used the average delay of the flights to compare with the speed.

```
PROC SQL;
CREATE TABLE Speed_AD AS
    SELECT DISTINCT(P.tailnum), P.type, max(P.speed) AS Speed, engines, model,
avg(F.arr_delay) AS Arrival_Delay, avg(F.dep_delay) AS Departure_Delay
    FROM planes_d.planes AS P
        INNER JOIN planes_d.flights AS F
            ON P.tailnum=F.tailnum
        INNER JOIN planes_d.airlines AS A
            ON F.carrier=A.carrier
    WHERE P.speed IS NOT NULL
    GROUP BY P.tailnum
    ORDER BY 4 DESC, 5 DESC;
    QUIT;
   RUN;
```

The following code was used to identify average arrival and departure delays between various origin airports and destinations. This was used to compare the delays with the distances.

```
PROC SQL outobs=100;
CREATE TABLE Dist_delv2 AS
    SELECT DISTINCT P.tailnum, F.carrier, F.origin, F.dest, P.seats AS total_seats, F.distance,
avg(F.arr_delay) AS Arrival_Delay, avg(F.dep_delay) AS Departure_Delay
    FROM planes_d.planes AS P, planes_d.flights AS F
        WHERE P.tailnum=F.tailnum
    GROUP BY P.tailnum
    ORDER BY 7 DESC;
QUIT;
RUN;
```

FLIGHTS DATA SET

In order to analyze the information with more ease, we ran small queries and created small tables.

Here is an example:

```
PROC SQL outobs  = 808;
CREATE TABLE top10_tail2 AS
SELECT tailnum, sum(dep_delay) as Dep_Delay
    FROM g_a.flights
    WHERE dep_delay > 0
    GROUP BY tailnum
    ORDER BY 2 DESC
;
QUIT;
```

This query filter the information to have an output of the 10% most delayed flights but the tailnum key to later analyze it with the planes date set.

Then, we created this table which already has only the top 10% to analyze the data un planes, like the airplane size (seats) and the its delayed or the manufactured and its year built.

```
PROC SQL;
CREATE TABLE g_a.y_delsum_man_nbr_mod_en_seats AS
SELECT year, sum(delay) as delay, manufacturer, count(manufacturer) as nbr_plains, model, engine, seats
    FROM new
    GROUP BY year, manufacturer, model, engine, seats
    ORDER BY delay DESC
```

```
;
QUIT;
```

We also, added the number of plains manufactured by each manufactured the accumulated delay.

```
PROC SQL;
CREATE TABLE g_a.y_delsum_man_nbr_mod_en_seats AS
SELECT year, sum(delay) as delay, manufacturer, count(manufacturer) as nbr_plains, model,
engine, seats
    FROM new
    GROUP BY year, manufacturer, model, engine, seats
    ORDER BY delay DESC
;
QUIT;
```

Moreover, we made queries to get useful information like the worst day to travel, worst hour, and worst month, all in terms of avg delay.

```
PROC SQL;
CREATE TABLE g_a.month_nbrFlights_arravg_deptavg AS
SELECT dn.month, dn.nbr_flight, ad.arr_avg_del as arr_avg_del, avg_del as dep_avg_del
    FROM dep_nbr_flights_month as dn, dep_avg_flights_months as dd,
arr_avg_flights_months as ad
    WHERE dn.month = dd.month
        AND ad.month = dd.month
    ORDER BY dep_avg_del DESC
;
QUIT;
```