Group Assignment Business Reporting Tools

NYC Flights data

Technical Report

We saved the different sources of data using the following piece of code, in which we specified the file location in our computer:

```
libname flights "File\Location";
```

To organize the data from the different sources and to use it to graph it in tableau we created the following additional tables:

- epavgdelays_airline_airport
- epdelays_per_airline
- epdelays_airline_airport
- weather_in_flightsperhout
- pdelays_per_airport
- summary_delays(2)
- epdelays_per_airline_size
- epdelaysavg_per_airline_size
- arrdelays_all_airports
- elayed_arr_flights
- nbr_flights_origin
- nbr_flights_per_airline
- numberofseats
- summary_delays

We created these tables using the option INNER JOIN…ON and selecting the variables that we wanted to explore. We also gave a name to each table and saved them under the created library "flights" using this piece of code:

```
create table flights.NameOfTable as
```

To create some of these joins we had to create temporal tables that helped us making the query fast. For example:

```
/*weather_in_flightsperhout:*/
proc sql;
create table flights.weather_in_flightsperhour as
```

```
select
w.origin as origin,
w.month as month,
w.day as day,
w.hour as hour,
avg(w.humid) as humid,
avg(w.temp) as temp,
avg(wind_speed) as wind_temp,
avg(wind_gust) as wind_gust,
avg(w.precip) as precip,
avg(w.pressure) as pressure,
avg(w.visib) as visib,
avg(f.dep_delay) as dep_delay,
avg(f.air_time) as air_time,
avg(f.distance) as distance,
a.lat as latitude,
a.lon as longitude
from flights.weather w
inner join flights.flightsperhour f
on w.hour=f.hour
and w.day=f.day
and w.month=f.month
and w.year=f.year
and w.origin=f.origin
inner join flights.airports a
on a.faa=f.origin
group by w.hour, w.day, w.month, w.year, w.origin
;
quit;
run;
```

In some SQL codes we did a modification to some variables and categorized them to help us with the analysis.

```
/*epavgdelays_airline_airport:*/
proc sql;
create table flights.epdelays_per_airport as
select
case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
a.name as airportsname,
f.origin as airports,
avg(f.dep_delay) as depdelays,
sum(f.dep_delay>0)/count(*) as percentofdelayedflights
from flights.flights f
inner join flights.airports a
on a.faa=f.origin
inner join flights.planes p
on f.tailnum=p.tailnum
```

```
group by 1,2,3
order by 1,2
;
quit;
run;
```

We focused our analysis in the arrival delays to the airports of NYC and in particular cases we filtered the information (using a WHERE clause) to work with only the delayed flights (ignoring the flights on time). However, in some other cases, we used the entire table to calculate the percentages of delayed flights as in the code below:

```
/*epdelays_airline_airport:*/
proc sql;
create table flights.epdelays_airline_airport as
select t1.origin, t2.name, t2.carrier,
sum(t1.dep_delay>0)/count(*) as percentofdep_delayedflights,
count(*) as total_flights
from flights.flights as t1, flights.airlines as t2
where t1.carrier=t2.carrier
group by 1,2,3
order by 1,3 DESC
;
quit;
```

Finally, we used the created tables to create different connections in tableau.

Code

```
/*flights:*/
proc sql;
create table flights.delayed_arr_flights as
select *
from flights.flights
where arr_delay>0
;
quit;

/*epavgdelays_airline_airport:*/
proc sql;
create table flights.epdelays_per_airport as
select
case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
a.name as airportsname,
f.origin as airports,
avg(f.dep_delay) as depdelays,
sum(f.dep_delay>0)/count(*) as percentofdelayedflights
from flights.flights f
inner join flights.airports a
on a.faa=f.origin
inner join flights.planes p
on f.tailnum=p.tailnum
group by 1,2,3
order by 1,2
;
quit;
run;

/*epdelays_per_airline:*/
proc sql;
create table flights.epdelays_per_airline as
select
case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
f.origin as airport,
a.name as airline,
a.carrier,
round(avg(f.dep_delay),0.01) as avg_time_dep_delay,
round(sum(f.dep_delay>0)/count(*),0.01) as
percentofdelayedflights
from flights.airlines as a,
flights.flights as f
inner join flights.planes p
on f.tailnum=p.tailnum
where a.carrier=f.carrier
group by 1,2,3,4
order by 1,3 DESC
;
quit;

/*epdelays_airline_airport:*/
proc sql;
create table flights.epdelays_airline_airport as
select t1.origin, t2.name, t2.carrier,
sum(t1.dep_delay>0)/count(*) as percentofdep_delayedflights,
count(*) as total_flights
from flights.flights as t1, flights.airlines as t2
where t1.carrier=t2.carrier
group by 1,2,3
order by 1,3 DESC
;
quit;

/*weather_in_flightsperhout:*/
proc sql;
create table flights.weather_in_flightsperhour as
```

```
select
w.origin as origin,
w.month as month,
w.day as day,
w.hour as hour,
avg(w.humid) as humid,
avg(w.temp) as temp,
avg(wind_speed) as wind_temp,
avg(wind_gust) as wind_gust,
avg(w.precip) as precip,
avg(w.pressure) as pressure,
avg(w.visib) as visib,
avg(f.dep_delay) as dep_delay,
avg(f.air_time) as air_time,
avg(f.distance) as distance,
a.lat as latitude,
a.lon as longitude
from flights.weather w
inner join flights.flightsperhour f
on w.hour=f.hour
and w.day=f.day
and w.month=f.month
and w.year=f.year
and w.origin=f.origin
inner join flights.airports a
on a.faa=f.origin
group by w.hour, w.day, w.month, w.year, w.origin
;
quit;
run;

/*pdelays_per_airport:*/
proc sql;
create table flights.epdelays_per_airport as
select
case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
a.name as airportsname,
f.origin as airports,
avg(f.dep_delay) as depdelays,
sum(f.dep_delay>0)/count(*) as percentofdelayedflights
from flights.flights f
inner join flights.airports a
on a.faa=f.origin
inner join flights.planes p
on f.tailnum=p.tailnum
group by 1,2,3
order by 1,2
;
quit;
run;
```

```
/*epdelays_per_airline_size:*/
proc sql;
create table flights.epdelays_per_airline_size as
select case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
a.name as airline,
a.carrier,
round(sum(f.dep_delay>0)/count(*),0.01) as
percentofdep_delayedflights,
count(*) as total_flights
from flights.airlines as a,
flights.flights as f
inner join flights.planes p
on f.tailnum=p.tailnum
where a.carrier=f.carrier
group by 1,2,3
order by 1,3 DESC
;
quit;

/*epdelaysavg_per_airline_size:*/
proc sql;
create table flights.epdelaysavg_per_airline_size as
select case when p.engine in ('Turbo-fan','Turbo-jet') then 'Big
Flights'
     else 'Small Flights' end as engine,
a.name as airline, a.carrier, round(avg(f.dep_delay),0.01) as
avg_time_dep_delay
from flights.airlines as a, flights.flights as f, flights.planes
as p
where a.carrier=f.carrier and p.tailnum=f.tailnum
and f.dep_delay>0
group by 1, 2,3
order by 1, 4 DESC
;
quit;

/*elayed_arr_flights:*/
proc sql;
create table flights.delayed_arr_flights as
select *
from flights.flights
where arr_delay>0
;
quit;

/*nbr_flights_origin:*/
proc sql;
create table flights.nbr_flights_origin as
```

```
select origin, sum(dep_delay>0) as nbr_delay_dep_flights,
count(origin) as total_flights, (sum(dep_delay>0)/count(origin))
as perc_delays
from flights.flights
group by 1
order by 3 DESC
;
quit;

/*nbr_flights_per_airline:*/
proc sql;
create table flights.nbr_flights_per_airline as
select t1.name as airline, t1.carrier, count(t2.flight) as
total_flights, sum(t2.dep_delay>0 OR t2.arr_delay>0) as
delayed_flights,
      sum(t2.dep_delay<=0 AND t2.arr_delay<=0) as
punctual_flights
from flights.airlines as t1, flights.flights as t2
where t1.carrier=t2.carrier
group by 1,2
order by 1 DESC, 2 DESC, 3 DESC
;
quit;

/*summary_delays(2):*/
proc sql;
create table flights.summary_delays as
select t1.carrier, t1.name, t2.origin, t2.month, t2.day, t2.hour,
count(t2.flight) as nbr_flights, sum(dep_delay) as sum_dep_delay,
avg(dep_delay) as avg_dep_delay
from flights.flights as t2, flights.airlines as t1
where t1.carrier=t2.carrier
group by 1,2,3,4,5,6
;
quit;

/*arrdelays_all_airports:*/
proc sql;
create table flights.delays_all_airports as
select a.name as airport_name, a.alt, a.lon, sum(f.arr_delay) as
delay
from flights.flights as f, flights.airports as a
group by 1,2;
quit;

/*numberofseats:*/
proc sql;
create table flights.numberofseats1 as
select (case when (t3.seats<=50) then '1. 2-50 Seats'
                when (t3.seats<=150) then '2. 51-150 Seats'
                when (t3.seats<=250) then '3. 151-250 Seats'
                when (t3.seats<=350) then '4. 251-350 Seats'
                else '5. 351-450 Seats' end) as nbr_seats,
                count(t2.flight) as total_flights,
sum(t2.dep_delay>0 OR t2.arr_delay>0) as delayed_flights,
                (sum(t2.dep_delay>0 OR
t2.arr_delay>0))/count(t2.flight) as perc_delays FORMAT=PERCENT.4
from flights.flights as t2, flights.planes as t3
where t2.tailnum=t3.tailnum
group by 1
;
quit;

/*summary_delays:*/
proc sql;
create table flights.summary_delays as
select t1.carrier, t1.name, t2.origin, t2.month, t2.day, t2.hour,
count(t2.flight) as nbr_flights, sum(dep_delay) as sum_dep_delay,
avg(dep_delay) as avg_dep_delay
from flights.flights as t2, flights.airlines as t1
where t1.carrier=t2.carrier
group by 1,2,3,4,5,6
;
quit;
```