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**;