



Startup “Flying meals”

Interest on:

- “After hours” flights.
 - Airport/City offer
- Delayed flights.
 - Complimentary from the airline

Goals after Data Analysis:

- Sell “Flying Meals” to the 4 Airlines* with more than 30% of their flights delayed and afterhours. Spirit Airlines Co. (NK), Frontier Airlines Inc. (F9), JetBlue Airways (B6) and United Air Lines Inc. (UA). They can offer “Flying Meals” boxes to all their passengers affected as complimentary.
- Install a food truck in Los Angeles International Airport (LAX), Chicago O'Hare International Airport (ORD) and Hartsfield-Jackson Atlanta International Airport (ATL) because they have the most afterhours flights and those cities are also affected with afterhours flights in all the others airports.
- Test stand at Chicago O' Hare Airport in collaboration with United Airlines. (ORD is the largest UA hub in terms of passengers carried and the number of departures**)

* NK: Spirit Airlines Co. Headquartered at Miami.
F9: Frontier Airlines Inc. Headquartered at Denver.
B6: JetBlue Airways. Headquartered at New York.
UA: United Air Lines Inc. Headquartered at Chicago.

Python (Google Colab)

```
✓ [1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import mpl_toolkits
%matplotlib inline
from google.colab import drive
from google.colab import files
pd.set_option("display.max_columns", None)
```

```
✓ [2] drive.mount('/content/gdrive')
4s
Mounted at /content/gdrive
```

```
✓ [3] from google.colab import drive
4s drive.mount('/content/drive')
Mounted at /content/drive
```

```
✓ [4] #data = pd.read_csv("/Users/Amparo/Desktop/DataAnalysis/BI_Intermediate/PowerBI/Project_BI_Advanced/flights.csv", sep=',', header=0)
0s #path = "Users/Amparo/Desktop/DataAnalysis/BI_Intermediate/PowerBI/Project_BI_Advanced/flights.csv"
```

```
✓ [5] flights = "/content/gdrive/MyDrive/Colab Notebooks/Project_BI_Advanced/flights.csv"
0s
```

```
✓ [6] flights = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Project_BI_Advanced/flights.csv', sep=',', header=0)
34s
```

```
✓ [7] flights
0s
```

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	WHEELS_OFF	SC
0	2015	1	1	4	AS	98	N407AS	ANC	SEA	5	2354.0	-11.0	21.0	15.0	
1	2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	10	2.0	-8.0	12.0	14.0	
2	2015	1	1	4	US	840	N171US	SFO	CLT	20	18.0	-2.0	16.0	34.0	
3	2015	1	1	4	AA	258	N3HYAA	LAX	MIA	20	15.0	-5.0	15.0	30.0	
4	2015	1	1	4	AS	135	N527AS	SEA	ANC	25	24.0	-1.0	11.0	35.0	
...	
5819074	2015	12	31	4	B6	688	N657JB	LAX	BOS	2359	2355.0	-4.0	22.0	17.0	



Total amount of values

df.describe() helps to make a fast analysis of the data

✓ 7s [9] flights.describe()

	YEAR	MONTH	DAY	DAY_OF_WEEK	FLIGHT_NUMBER	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	WHEELS_OFF	SCHEDULED_TIME	ELAPSED_TIME
count	5819079.0	5.819079e+06	5.819079e+06	5.819079e+06	5.819079e+06	5.819079e+06	5.732926e+06	5.732926e+06	5.730032e+06	5.730032e+06	5.819073e+06	5.714008e+06
mean	2015.0	6.524085e+00	1.570459e+01	3.926941e+00	2.173093e+03	1.329602e+03	1.335204e+03	9.370158e+00	1.607166e+01	1.357171e+03	1.416859e+02	1.370062e+02
std	0.0	3.405137e+00	8.783425e+00	1.988845e+00	1.757064e+03	4.837518e+02	4.964233e+02	3.708094e+01	8.895574e+00	4.980094e+02	7.521058e+01	7.421107e+01
min	2015.0	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	-8.200000e+01	1.000000e+00	1.000000e+00	1.800000e+01	1.400000e+01
25%	2015.0	4.000000e+00	8.000000e+00	2.000000e+00	7.300000e+02	9.170000e+02	9.210000e+02	-5.000000e+00	1.100000e+01	9.350000e+02	8.500000e+01	8.200000e+01
50%	2015.0	7.000000e+00	1.600000e+01	4.000000e+00	1.690000e+03	1.325000e+03	1.330000e+03	-2.000000e+00	1.400000e+01	1.343000e+03	1.230000e+02	1.180000e+02
75%	2015.0	9.000000e+00	2.300000e+01	6.000000e+00	3.230000e+03	1.730000e+03	1.740000e+03	7.000000e+00	1.900000e+01	1.754000e+03	1.730000e+02	1.680000e+02
max	2015.0	1.200000e+01	3.100000e+01	7.000000e+00	9.855000e+03	2.359000e+03	2.400000e+03	1.988000e+03	2.250000e+02	2.400000e+03	7.180000e+02	7.660000e+02

✓ 7s [9] flights.describe()

ED_TIME	ELAPSED_TIME	AIR_TIME	DISTANCE	WHEELS_ON	TAXI_IN	SCHEDULED_ARRIVAL	ARRIVAL_TIME	ARRIVAL_DELAY	DIVERTED	CANCELLED	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY
073e+06	5.714008e+06	5.714008e+06	5.819079e+06	5.726566e+06	5.726566e+06	5.819079e+06	5.726566e+06	5.714008e+06	5.819079e+06	5.819079e+06	1.063439e+06	1.063439e+06	1.063439e+06	1.063439e+06	1.063439e+06
859e+02	1.370062e+02	1.135116e+02	8.223565e+02	1.471469e+03	7.434971e+00	1.493808e+03	1.476491e+03	4.407057e+00	2.609863e-03	1.544643e-02	1.348057e+01	7.615387e-02	1.896955e+01	2.347284e+01	2.915290e+00
058e+01	7.421107e+01	7.223082e+01	6.077843e+02	5.221879e+02	5.638548e+00	5.071647e+02	5.263197e+02	3.927130e+01	5.102012e-02	1.233201e-01	2.800368e+01	2.143460e+00	4.816164e+01	4.319702e+01	2.043334e+01
000e+01	1.400000e+01	7.000000e+00	2.100000e+01	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	-8.700000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
000e+01	8.200000e+01	6.000000e+01	3.730000e+02	1.054000e+03	4.000000e+00	1.110000e+03	1.059000e+03	-1.300000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
000e+02	1.180000e+02	9.400000e+01	6.470000e+02	1.509000e+03	6.000000e+00	1.520000e+03	1.512000e+03	-5.000000e+00	0.000000e+00	0.000000e+00	2.000000e+00	0.000000e+00	2.000000e+00	3.000000e+00	0.000000e+00
000e+02	1.680000e+02	1.440000e+02	1.062000e+03	1.911000e+03	9.000000e+00	1.918000e+03	1.917000e+03	8.000000e+00	0.000000e+00	0.000000e+00	1.800000e+01	0.000000e+00	1.900000e+01	2.900000e+01	0.000000e+00
000e+02	7.660000e+02	6.900000e+02	4.983000e+03	2.400000e+03	2.480000e+02	2.400000e+03	2.400000e+03	1.971000e+03	1.000000e+00	1.000000e+00	1.134000e+03	5.730000e+02	1.971000e+03	1.331000e+03	1.211000e+03

```
✓ [31] #Checking null values in each column
5s flights.isnull().sum()
```

```
YEAR                0
MONTH               0
DAY                0
DAY_OF_WEEK         0
AIRLINE             0
FLIGHT_NUMBER       0
TAIL_NUMBER        14721
ORIGIN_AIRPORT      0
DESTINATION_AIRPORT 0
SCHEDULED_DEPARTURE 0
DEPARTURE_TIME      86153
DEPARTURE_DELAY     86153
TAXI_OUT            89047
WHEELS_OFF          89047
SCHEDULED_TIME      6
ELAPSED_TIME        105071
AIR_TIME            105071
DISTANCE            0
WHEELS_ON           92513
TAXI_IN             92513
SCHEDULED_ARRIVAL   0
ARRIVAL_TIME        92513
ARRIVAL_DELAY       105071
DIVERTED            0
CANCELLED           0
CANCELLATION_REASON 5729195
AIR_SYSTEM_DELAY    4755640
SECURITY_DELAY      4755640
AIRLINE_DELAY       4755640
LATE_AIRCRAFT_DELAY 4755640
WEATHER_DELAY       4755640
dtype: int64
```

The “null values” in departure and arrival delays are because of Cancelled or Diverted flights

```
✓ [32] #Checking number of flights Cancelled
0s flights['CANCELLED'].value_counts()
```

```
0    5729195
1      89884
Name: CANCELLED, dtype: int64
```

```
✓ [38] #Checking number of flights Diverted
1s flights['DIVERTED'].value_counts()
```

```
0    5803892
1      15187
Name: DIVERTED, dtype: int64
```

SQL (BigQuery)

Delayed flights

```
EJECUTAR GUARDAR COMPARTIR PROGRAMACIÓN MÁS
1 SELECT AIRLINE, COUNT(*) AS Total_arrival_delayed_flights
2 FROM (
3   SELECT AIRLINE,
4     CASE WHEN ARRIVAL_DELAY > 60 THEN 'Arrival_Delayed'
5         ELSE 'On_time'
6     END AS Delay
7   FROM ToGoBox.flights
8 ) f
9 WHERE Delay = 'Arrival_Delayed'
10 GROUP BY AIRLINE, Delay
11 order by Total_arrival_delayed_flights DESC
```

After hours flights

```
EJECUTAR GUARDAR COMPARTIR PROGRAMACIÓN MÁS
1
2 SELECT AIRLINE, COUNT(*) AS Total_after_hours_flights
3 FROM (
4   SELECT AIRLINE,
5     CASE WHEN SCHEDULED_DEPARTURE > 800 and SCHEDULED_DEPARTURE <2200 THEN 'Regular'
6         ELSE 'After_hours'
7     END AS Flight_time
8   FROM ToGoBox.flights
9 ) f
10 WHERE Flight_time = 'After_hours'
11 GROUP BY AIRLINE, Flight_time
12 order by Total_after_hours_flights DESC
13
14
15
16
```

Transform Data to .CSV and export to Google Colab

Delayed flights by Airline

All Airlines have delayed flights in a lower percent of 15%.
Non significant.

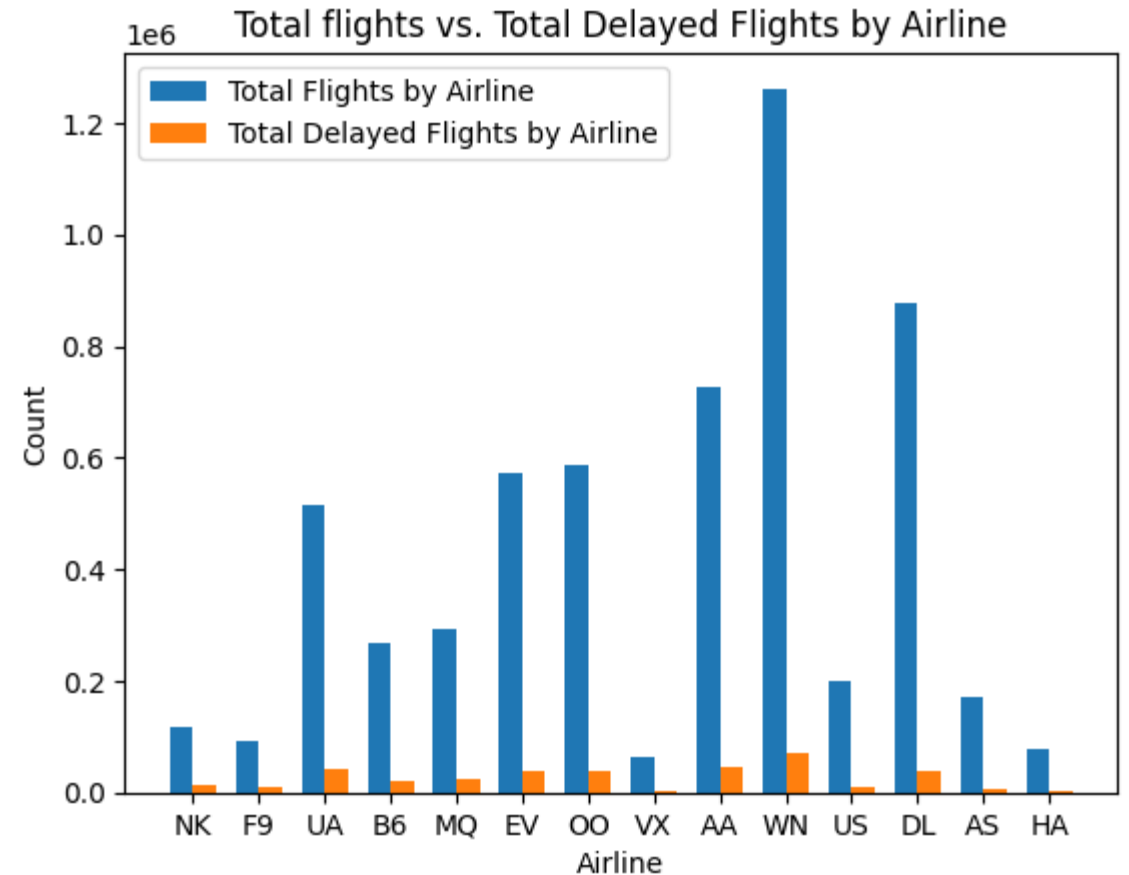
```
✓ 0s TotalANDdelayed = pd.merge(FlightsAirline, TotalDelayedAirline, on='AIRLINE')  
print(TotalANDdelayed)
```

	AIRLINE	Count	Delayed_Flights
0	WN	1261855	71344
1	DL	875881	39785
2	AA	725984	44572
3	OO	588353	38288
4	EV	571977	40046
5	UA	515723	43664
6	MQ	294632	22592
7	B6	267048	22147
8	US	198715	10121
9	AS	172521	5534
10	NK	117379	13079
11	F9	90836	8713
12	HA	76272	1139
13	VX	61903	3906

```
✓ 0s [15] # Calculate the percentage of delayed flights by airline  
TotalANDdelayed['Percent_Delayed'] = 100 * TotalANDdelayed['Delayed_Flights'] / TotalANDdelayed['Count']  
TotalANDdelayed = TotalANDdelayed.sort_values(by='Percent_Delayed', ascending=False)  
  
print(TotalANDdelayed[['AIRLINE', 'Percent_Delayed']])
```

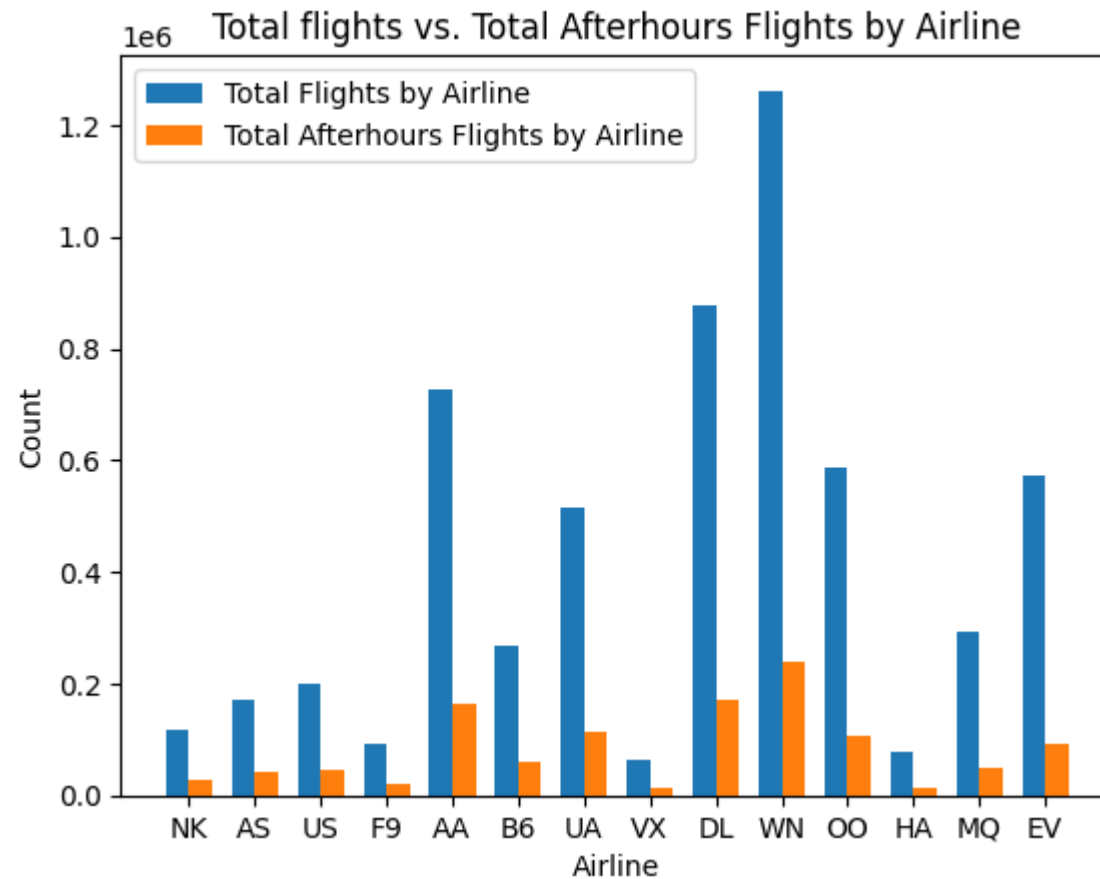
	AIRLINE	Percent_Delayed
10	NK	11.142538
11	F9	9.592012
5	UA	8.466561
7	B6	8.293266
6	MQ	7.667870
4	EV	7.001330
3	OO	6.507658
13	VX	6.309872
2	AA	6.139529
0	WN	5.653898
8	US	5.093224
1	DL	4.542284
9	AS	3.207725
12	HA	1.493340

```
[12] # Set the width of each bar  
bar_width = 0.35  
  
# Create an array of indices for each airline  
indices = np.arange(len(TotalANDdelayed['AIRLINE']))  
  
# Create the bars for each column  
fig, ax = plt.subplots()  
ax.bar(indices - bar_width/2, TotalANDdelayed['Count'], width=bar_width, label='Total Flights by Airline')  
ax.bar(indices + bar_width/2, TotalANDdelayed['Delayed_Flights'], width=bar_width, label='Total Delayed Flights by Airline')  
  
# Set the axis labels and chart title  
ax.set_xlabel('Airline')  
ax.set_ylabel('Count')  
ax.set_title('Total Flights vs. Total Delayed Flights by Airline')  
  
# Set the x-axis tick labels to the airline names  
ax.set_xticks(indices)  
ax.set_xticklabels(TotalANDdelayed['AIRLINE'])  
  
# Display the legend and show the chart  
ax.legend()  
plt.show()
```



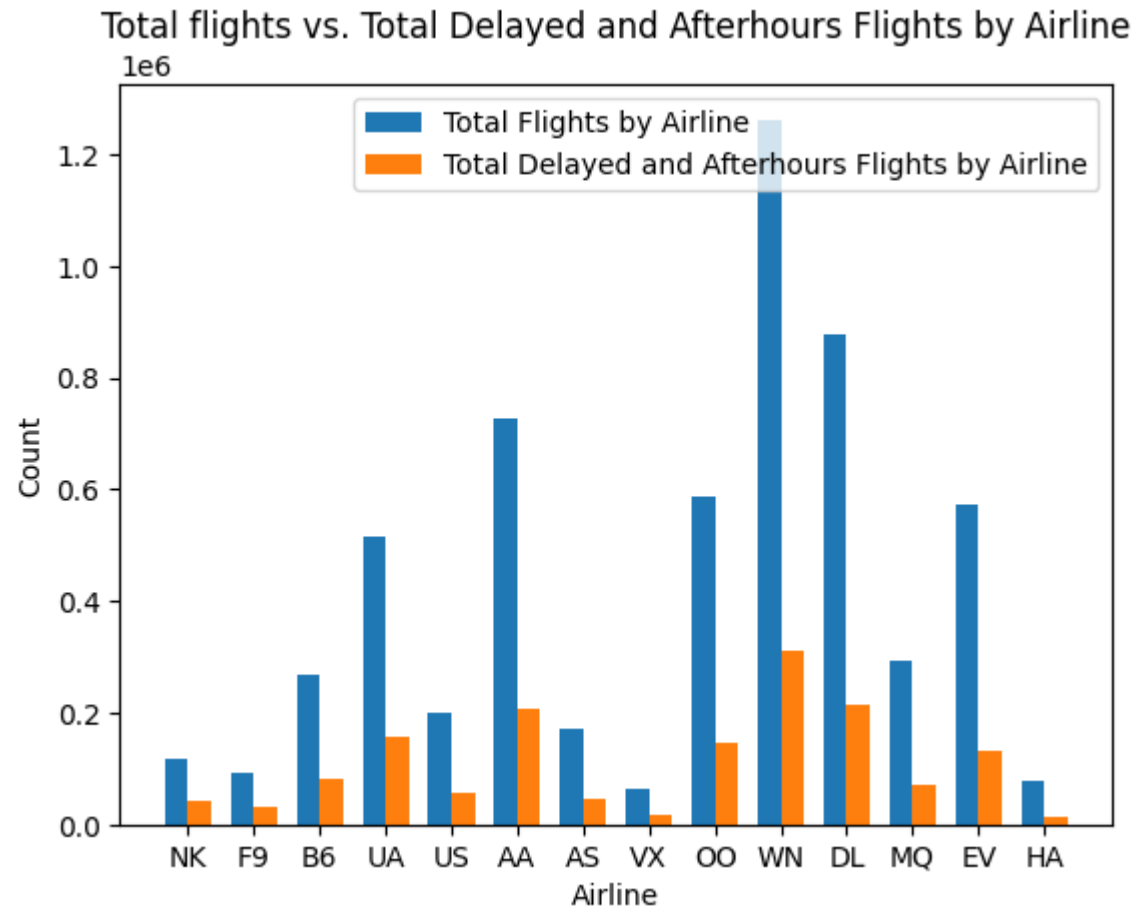
After-hours flights by Airline

All Airlines have after-hours flights in a lower percent of 25%.
Non significant.



Delayed and after-hours flights by Airline

Just delayed or after-hours flights by Airline are non significant, but both together are.



NK 35.10 %

F9 33.08 %

B6 30.55 %

UA 30.38 %

Sell the product to this 4 Airlines.

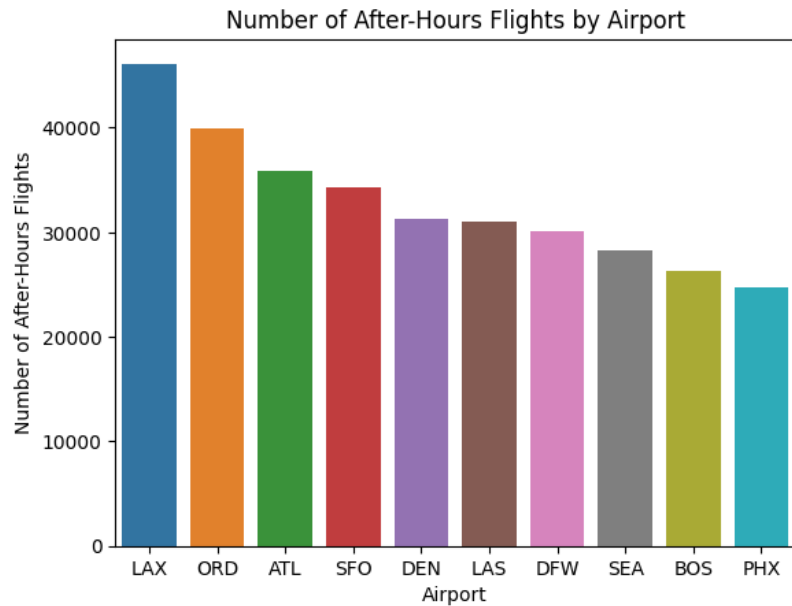
After-hours flights by Airport and City

```
[5] # Top 10 airports with after-hours flights
top10_airport_afterhours = AfterHoursAirport.head(10)['ORIGIN_AIRPORT'].tolist()

# Top 10 cities
AfterHoursAirport_top10 = AfterHoursAirport[AfterHoursAirport['ORIGIN_AIRPORT'].isin(top10_airport_afterhours)]

# plot the bar chart
sns.barplot(data=AfterHoursAirport_top10, x='ORIGIN_AIRPORT', y='Total_after_hours_flights')
plt.title('Number of After-Hours Flights by Airport')
plt.xlabel('Airport')
plt.ylabel('Number of After-Hours Flights')

plt.show()
```

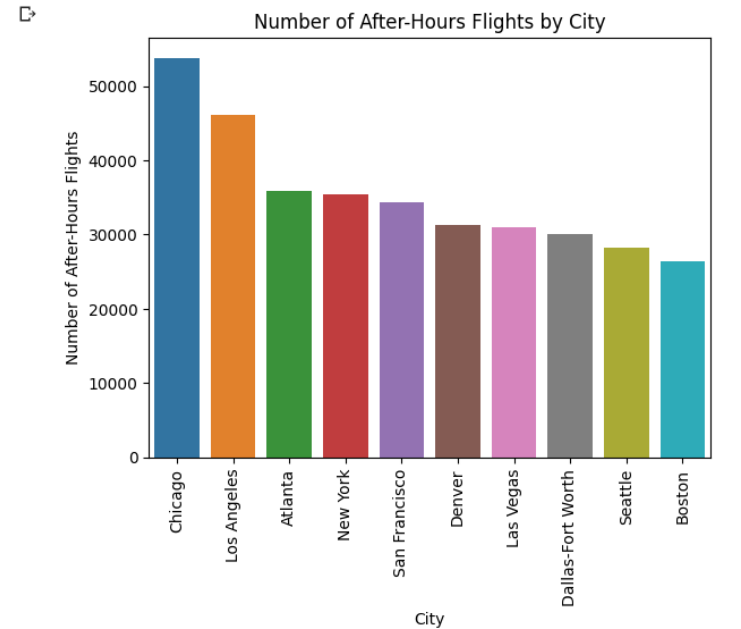


Top 3 Cities:

- Chicago
- Los Angeles
- Atlanta

Top 3 Airports:

- LAX: Los Angeles International Airport
- ORD: Chicago O'Hare International Airport
- ATL: Hartsfield-Jackson Atlanta International Airport



Sell the product in LAX, ORD and ATL

ORD: Chicago O'Hare International Airport



What's next?

- When perishable products, analyze the amount of afterhours flights per day, to reduce the waste.
- App to order personalized boxes.