



# Tecnológico de Monterrey

**TI - Evidencia 2 - Analítica descriptiva  
(Integración de datos a través de modelo  
entidad-relación) y el compromiso ético y  
ciudadano**

**Escuela de Negocios- Manipulación de  
Datos (Gpo 201)**

**Andres Posada Sanchez Cobiza**

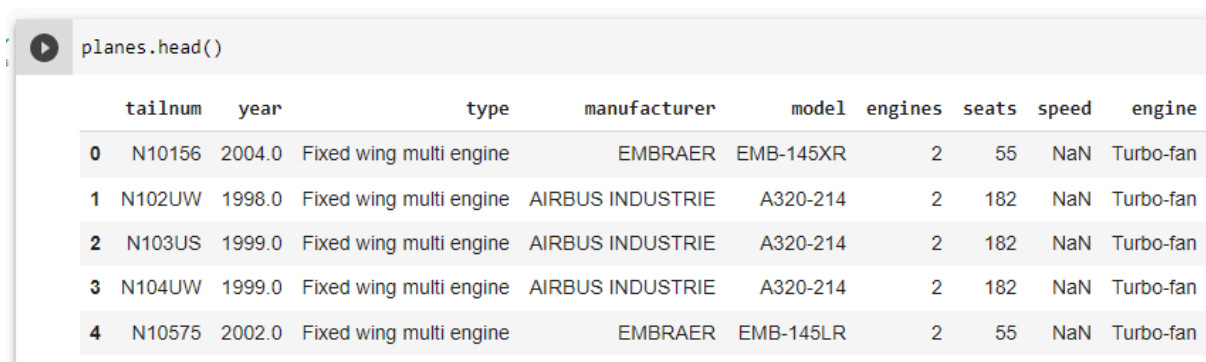
**A01382640**

04 de mayo de 2023

In this report I will analyze the opportunities that American Airlines has to better their competitive positioning, based on Data Analysis conducted with Google Colaboratory.

## Modelo Entidad-Relación

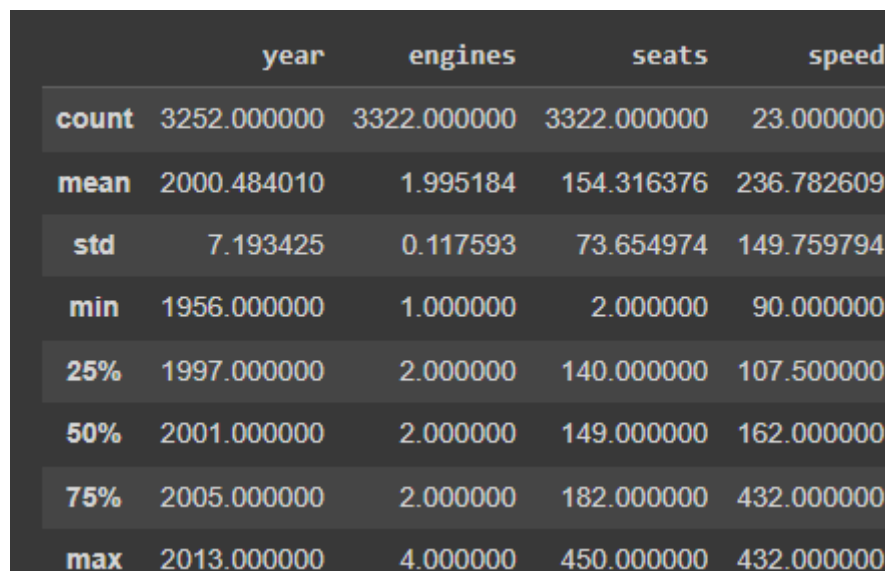
Firstly, using libraries from nycflights13 we can display several data frames which can give us helpful insights regarding flights, weather, carriers, airports and planes. The following images represent such data for the planes and weather databases, as well as the descriptive data from the fore mentioned datasets, obtained using `planes.describe()`.



```
planes.head()
```

	tailnum	year	type	manufacturer	model	engines	seats	speed	engine
0	N10156	2004.0	Fixed wing multi engine	EMBRAER	EMB-145XR	2	55	NaN	Turbo-fan
1	N102UW	1998.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
2	N103US	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
3	N104UW	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan
4	N10575	2002.0	Fixed wing multi engine	EMBRAER	EMB-145LR	2	55	NaN	Turbo-fan

first rows of the planes database using `planes.head()`



	year	engines	seats	speed
count	3252.000000	3322.000000	3322.000000	23.000000
mean	2000.484010	1.995184	154.316376	236.782609
std	7.193425	0.117593	73.654974	149.759794
min	1956.000000	1.000000	2.000000	90.000000
25%	1997.000000	2.000000	140.000000	107.500000
50%	2001.000000	2.000000	149.000000	162.000000
75%	2005.000000	2.000000	182.000000	432.000000
max	2013.000000	4.000000	450.000000	432.000000

Descriptive Data regarding Data Frame planes

	origin	year	month	day	hour	temp	dewp	humid	wind_dir	wind_speed	wind_gust	precip	pressure	visib
0	EWB	2013	1	1	1	39.02	26.06	59.37	270.0	10.35702	NaN	0.0	1012.0	10.0
1	EWB	2013	1	1	2	39.02	26.96	61.63	250.0	8.05546	NaN	0.0	1012.3	10.0
2	EWB	2013	1	1	3	39.02	28.04	64.43	240.0	11.50780	NaN	0.0	1012.5	10.0
3	EWB	2013	1	1	4	39.92	28.04	62.21	250.0	12.65858	NaN	0.0	1012.2	10.0
4	EWB	2013	1	1	5	39.02	28.04	64.43	260.0	12.65858	NaN	0.0	1011.9	10.0

first rows of the weather database using `weather.head()`

	year	month	day	hour	temp	dewp	humid	wind_dir
count	26115.0	26115.000000	26115.000000	26115.000000	26114.000000	26114.000000	26114.000000	25655.000000
mean	2013.0	6.503733	15.675321	11.490791	55.260392	41.439985	62.530059	199.761060
std	0.0	3.438328	8.762177	6.912423	17.787852	19.386236	19.395918	107.306847
min	2013.0	1.000000	1.000000	0.000000	10.940000	-9.940000	12.740000	0.000000
25%	2013.0	4.000000	8.000000	6.000000	39.920000	26.060000	47.050000	120.000000
50%	2013.0	7.000000	16.000000	11.000000	55.400000	42.080000	61.790000	220.000000
75%	2013.0	9.000000	23.000000	17.000000	69.980000	57.920000	78.790000	290.000000
max	2013.0	12.000000	31.000000	23.000000	100.040000	78.080000	100.000000	360.000000

Descriptive Data regarding Data Frame weather

Afterwards, the relevant information can be filtered. For this study case we need to know for each flight the airline, origin and destination. This can be done with the following line:

```
info_flights=flights[["carrier","origin","dest"]]
```

And then printing the result, which displays the following:

	carrier	origin	dest
0	UA	EWR	IAH
1	UA	LGA	IAH
2	AA	JFK	MIA
3	B6	JFK	BQN
4	DL	LGA	ATL
...	...	...	...
336771	9E	JFK	DCA
336772	9E	LGA	SYR
336773	MQ	LGA	BNA
336774	MQ	LGA	CLE
336775	MQ	LGA	RDU
336776 rows x 3 columns			

Now, to know the full name of each airline, we can get this information from the data frame “airlines”, and join it to the existing data using:

```
##merge the previous table to the data from caarriers
flightswtcarrier=pd.merge(info_flights, airlines, how="left", on
= "carrier")

##rearrange columns for carrier and name to be to the right
rearranged_info=flightswtcarrier[["origin","dest","carrier","name
"]]
```

which displays:

	origin	dest	carrier	name
0	EWR	IAH	UA	United Air Lines Inc.
1	LGA	IAH	UA	United Air Lines Inc.
2	JFK	MIA	AA	American Airlines Inc.
3	JFK	BQN	B6	JetBlue Airways
4	LGA	ATL	DL	Delta Air Lines Inc.
...	...	...	...	...
336771	JFK	DCA	9E	Endeavor Air Inc.
336772	LGA	SYR	9E	Endeavor Air Inc.
336773	LGA	BNA	MQ	Envoy Air
336774	LGA	CLE	MQ	Envoy Air
336775	LGA	RDU	MQ	Envoy Air
336776 rows x 4 columns				

With this new set of data, we can identify the most sought after destinations, by showing the 10 destinations with the most flights. Firstly, we need to add the name to count the times each destination is repeated, this can be done by using

```
populardest=rearranged_info.groupby("dest")["dest"].count().rename("count flights")
```

This also gives the new column with the times that each destination is repeated the name “count flights”. After turning the data back into a frame, we can rearrange the data frame so that the highest values on “count flights” are shown first, and display the first 10 values.

```
pdf=populardest.to_frame()
sortedpopulardest=pdf.sort_values(by=["count flights"],ascending=False)
sortedpopulardest.head(10)
```

The output obtained is the following dataframe

count flights	
dest	
ORD	17283
ATL	17215
LAX	16174
BOS	15508
MCO	14082
CLT	14064
SFO	13331
FLL	12055
MIA	11728
DCA	9705

With the data collected, we can conclude that ORD, ATL, LAX, and the rest of the list, are the destinations with the most flights as a destination from NYC. This was to be expected since these airports have the most air traffic in the USA and some of them are also some of the most populated cities in the country. To better the competitive positioning that American Airlines has, they should make sure that they take advantage of the demand of these destinations in particular.

Another valuable insight that we could obtain from our initial dataframes, is the amount of flights that each carrier has performed each month, to identify the months in which more flights are performed as well as which carriers are the most used

during said months. This can be done as follows:

```
[21] fgf=flights.groupby(["carrier","month"])["month"].count().rename("count flights")

[24] fgfframe=fgf.to_frame()

fgfframe.sort_values(by=["count flights"],ascending=False)
```

carrier	month	count flights
UA	8	5124
	7	5066
	10	5060
	4	5047
B6	7	4984
...	...	...
YV	3	18
OO	11	5
	8	4
	6	2
	1	1

185 rows x 3 columns

Interestingly enough, United has the top 4 entries, having the most flights in the month of august, followed closely by July. It isn't only until the fifth entry in which B6 makes an entry, also with the month of July. It is fair to conclude with this analysis that the most popular airline is United, while the most popular months are July and August.

Lastly, it would be insightful to know how many planes are manufactured by each airline. This can prove helpful when tracing which are the most important industry providers, as well as the quantities that they are producing.

To do this, we start off by creating a list of the carriers and the tail numbers assigned to them. This was done using the lines

```
carrierbytailnumber_list = flights[["carrier","tailnum"]]
carrierbytailnumber_list
```

Which shows us the list:

	carrier	tailnum
0	UA	N14228
1	UA	N24211
2	AA	N619AA
3	B6	N804JB
4	DL	N668DN
...	...	...
336771	9E	NaN
336772	9E	NaN
336773	MQ	N535MQ
336774	MQ	N511MQ
336775	MQ	N839MQ
336776 rows x 2 columns		

Then we eliminate duplicate values by using function `.drop_duplicates()`, as in line

```
carrierbytailnumber_list =
carrierbytailnumber_list.drop_duplicates()
```

Which replaces the variable and brings down the number of entries from 336,776 rows to 4,067.

Then, We use the data from the dataframe “planes”, and merge it with the list we just generated, which will keep the tail number as key to join to the left the carrier information corresponding to each plane. To do this we require the pandas function `pd.merge` as in line

```
x_new = pd.merge(planes, carrierbytailnumber_list, how = "left",
on = "tailnum")
```

in this occasion we see the following table:



[32] x\_new

	tailnum	year	type	manufacturer	model	engines	seats	speed	engine	carrier
0	N10156	2004.0	Fixed wing multi engine	EMBRAER	EMB-145XR	2	55	NaN	Turbo-fan	EV
1	N102UW	1998.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan	US
2	N103US	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan	US
3	N104UW	1999.0	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NaN	Turbo-fan	US
4	N10575	2002.0	Fixed wing multi engine	EMBRAER	EMB-145LR	2	55	NaN	Turbo-fan	EV
...	...	...	...	...	...	...	...	...	...	...
3334	N997AT	2002.0	Fixed wing multi engine	BOEING	717-200	2	100	NaN	Turbo-fan	FL
3335	N997DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS AIRCRAFT CO	MD-88	2	142	NaN	Turbo-fan	DL
3336	N998AT	2002.0	Fixed wing multi engine	BOEING	717-200	2	100	NaN	Turbo-fan	FL
3337	N998DL	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88	2	142	NaN	Turbo-jet	DL
3338	N999DN	1992.0	Fixed wing multi engine	MCDONNELL DOUGLAS CORPORATION	MD-88	2	142	NaN	Turbo-jet	DL

3339 rows x 10 columns

To better understand the data displayed, before determining how many planes are assigned to each manufacturer, we will add the name of the manufacturer to its initials. which will be done by merging the foreshown table to the table in “airlines”, using the key factor “carrier”, so that each carrier has it’s whole name by each entry. We do this using line

```
x_new_2 = pd.merge(x_new, airlines, how = "left", on = "carrier")
```

Before creating our final table with the amount of planes using

```
Final_table =  
x_new_2.groupby(['manufacturer', 'name'])['name'].count()
```

and turning it into a dataframe for better visualization, which shows us:

[39] Final\_table.to\_frame()

manufacturer	name	
AGUSTA SPA	American Airlines Inc.	1
AIRBUS	Delta Air Lines Inc.	38
	Frontier Airlines Inc.	21
	Hawaiian Airlines Inc.	14
	JetBlue Airways	110
	US Airways Inc.	85
	United Air Lines Inc.	15
	Virgin America	53
AIRBUS INDUSTRIE	AirTran Airways Corporation	1
	Delta Air Lines Inc.	92
	Frontier Airlines Inc.	2
	JetBlue Airways	17
	US Airways Inc.	151
	United Air Lines Inc.	137
AMERICAN AIRCRAFT INC	American Airlines Inc.	2
AVIAT AIRCRAFT INC	American Airlines Inc.	1
AVIONS MARCEL DASSAULT	Mesa Airlines Inc.	1
BARKER JACK L	JetBlue Airways	1
BEECH	American Airlines Inc.	2
BELL	American Airlines Inc.	2

BOEING	AirTran Airways Corporation	115
	Alaska Airlines Inc.	84
	American Airlines Inc.	56
	Delta Air Lines Inc.	333
	Southwest Airlines Co.	580
	US Airways Inc.	25
	United Air Lines Inc.	446
BOMBARDIER INC	Endeavor Air Inc.	203
	ExpressJet Airlines Inc.	88
	Mesa Airlines Inc.	57
	SkyWest Airlines Inc.	28
CANADAIR	ExpressJet Airlines Inc.	9
CANADAIR LTD	Envoy Air	1
CESSNA	American Airlines Inc.	7
	Envoy Air	2
CIRRUS DESIGN CORP	JetBlue Airways	1
DEHAVILLAND	American Airlines Inc.	1
DOUGLAS	American Airlines Inc.	1
EMBRAER	ExpressJet Airlines Inc.	219
	JetBlue Airways	60
	US Airways Inc.	20
FRIEDEMANN JON	American Airlines Inc.	1
GULFSTREAM AEROSPACE	American Airlines Inc.	1

HURLEY JAMES LARRY	American Airlines Inc.	1
JOHN G HESS	AirTran Airways Corporation	1
KILDALL GARY	American Airlines Inc.	1
LAMBERT RICHARD	American Airlines Inc.	1
LEARJET INC	American Airlines Inc.	1
LEBLANC GLENN T	American Airlines Inc.	1
MARZ BARRY	American Airlines Inc.	1
MCDONNELL DOUGLAS	American Airlines Inc.	81

Although some manufacturers have dominance on some airlines, it is clear that the most dominant airplane manufacturer is Boeing, with their staggering amount of planes produced for United and Southwest in particular.

## Data Visualization

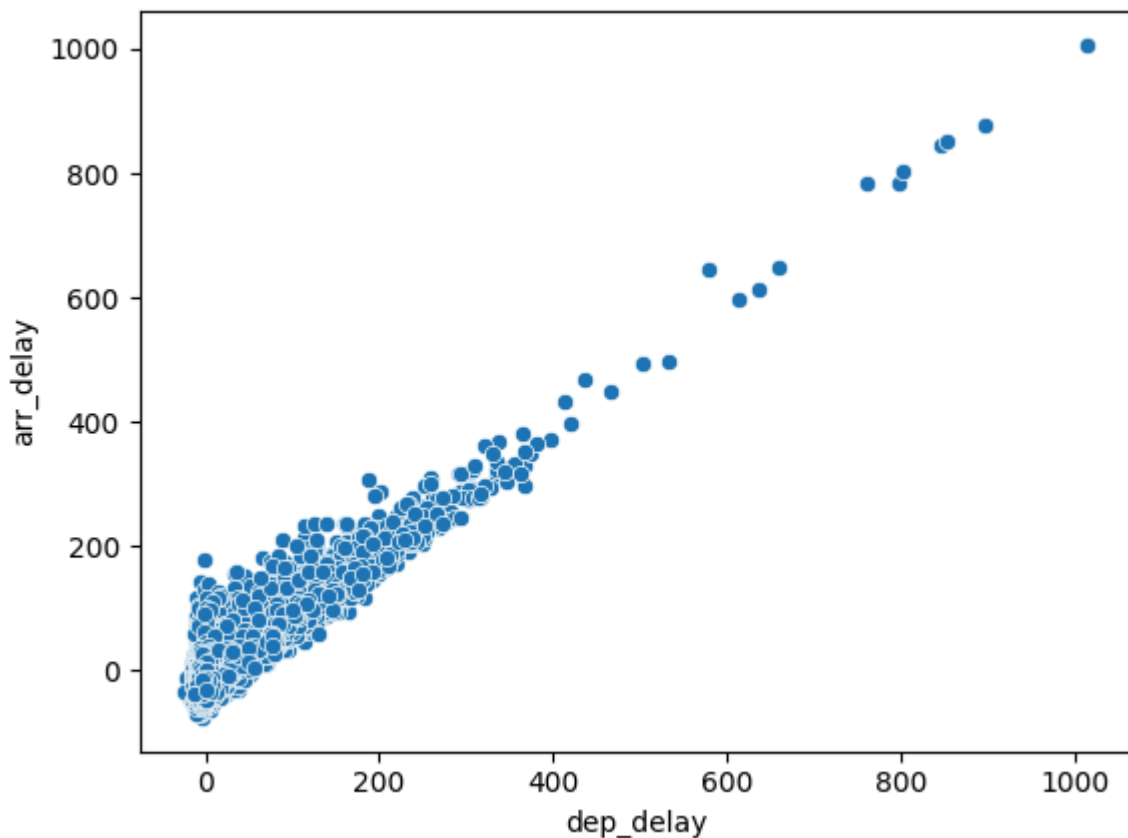
Our first task in the data visualization process, is to identify the relation between delays in time of arrival and delay in time of departure. To do this, we generate a scatter graph to show these two variables in the two axes. Firstly we need to import the required libraries, in this case seaborn as sns. Then, using the “flights” data frame, and filtering the American Airlines flights using

```
aaflights=flights[flights["carrier"]=="AA"]
```

we can use the functions:

```
sns.scatterplot(data=aaflights, x="dep_delay", y="arr_delay")
```

to see the graph generated:



Seeing the ascending straight line trend from left to right, we can see that the delay in departure and arrival are directly correlated. This could be inferred, although another interesting fact to observe is that there are several flights that arrived with a delay after departing with no delay, while the flights which departed with delay almost always arrived with a delay.

Analyzing airport weather trends can help predict the behavior in future occasions and is a tool that is used every day.

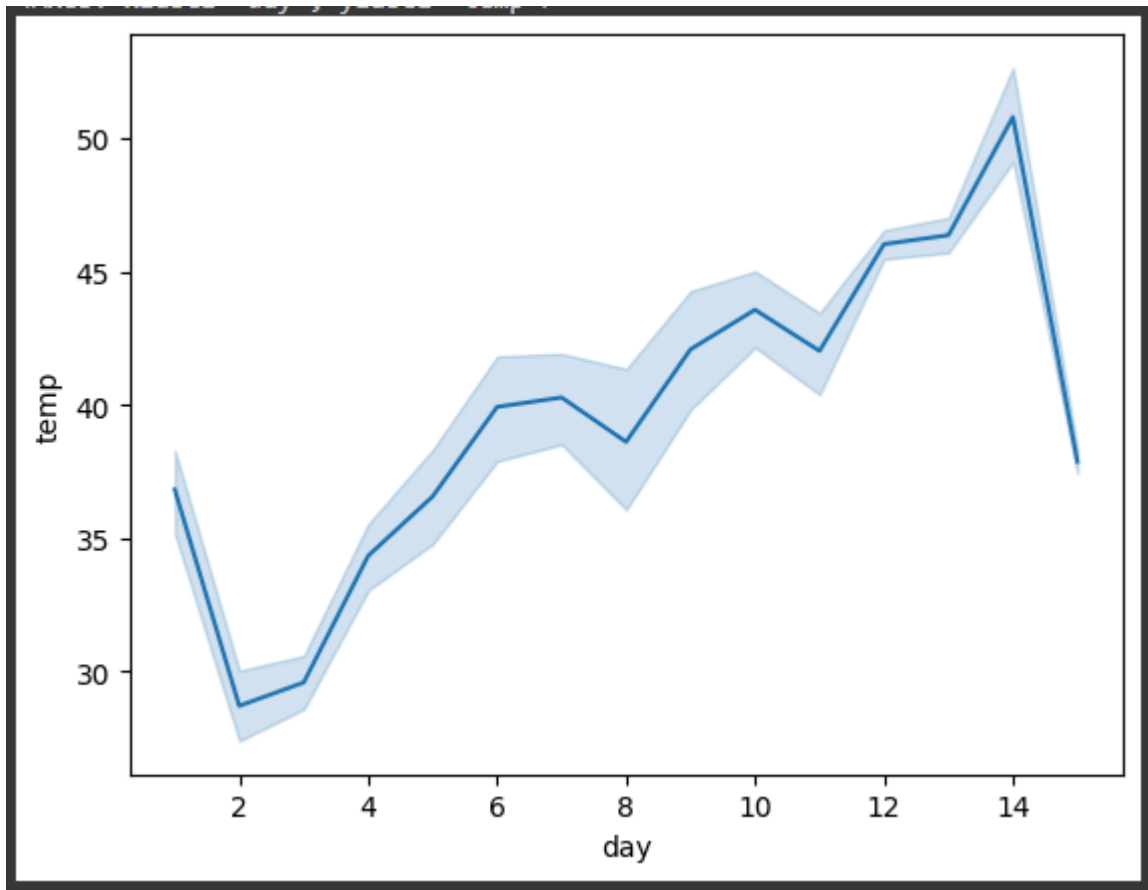
To make use of the data required to see the temperature of the first 15 days of the year departing EWR airport, a linear graph is the most helpful, and it can be done by first, filtering the days and airport selected using

```
Df_new = weather[weather["origin"] == "EWR"]  
Df_new = Df_new[Df_new["day"] <= 15]  
Df_new = Df_new[Df_new["month"] == 1]
```

using the line graph function

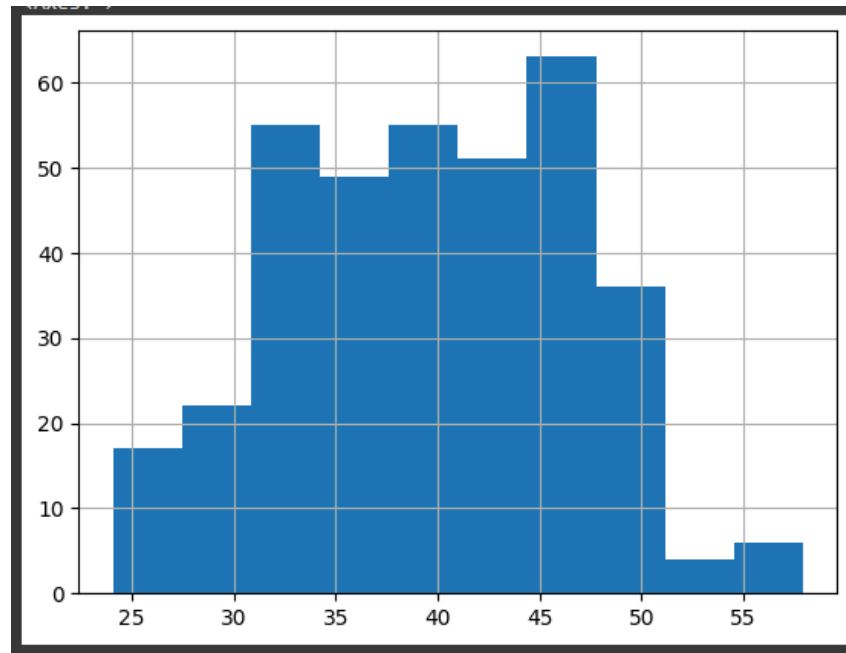
```
sns.lineplot(data= Df_new, x="day", y="temp")
```

finally displaying:



The linear graph allows us to see that the temperature is increasing gradually, which is the trend when winter is dissipating, while on the 14 there was a sudden drop in temperature. This is a surprising discovery since one would expect the trend of increasing temperature to continue, but this now makes us dive deeper into the reasons for this sudden drop, which could be a storm or a strong wind current.

To have a more general idea about the temperatures of this season, the same can be done using a histogram for the days in our new database. This can be done simply by using `Df_new["temp"].hist()` which displays the following:



we can observe through the graph that the most common temperatures are those near 45 degrees, accumulating up to more than 60 hours in which the temperature fell in these ranges. Notably, this is a tool that can prove to be misleading, since it just means that it was the most stable one, and with the help of the linear graph we made earlier, we can see that the temperature really is not that stable, and it changes day by day.

As our last task, we are instructed to find the 10 airlines with the most flights that departed NYC in 2013, and show it in a bar graph as well as in a pie chart. The information can be retrieved from the dataframe flights. We then introduce the following lines of code, firstly to form columns matching the names of the carriers to their number of flights assigned to each carrier.

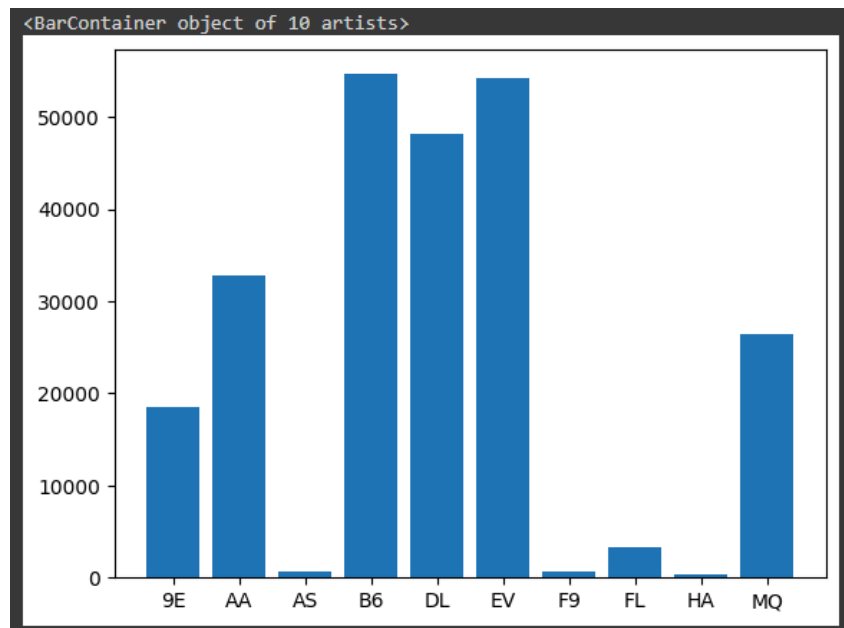
```
New_df =
flights.groupby(['carrier'])['carrier'].count().rename("count flights")
New_df = New_df.to_frame()
```

Arranging them from the carrier with the most departed flights to the one with the least, and filtering the top 10 results.

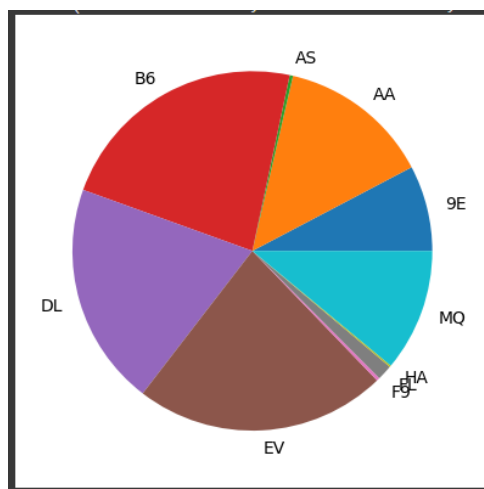
```
New_df.sort_values(by=['count flights'], ascending=False)
New_df_10 = New_df[0:10]
```

and finally displaying the data in bars and the pie chart.

```
plt.bar(New_df_10.index, New_df_10['count flights'])
```



```
fig, ax = plt.subplots()
ax.pie( New_df_10["count flights"],
labels=New_df_10.index)
```



These tables provide valuable insights similar to the ones done in our first task, where we can observe, now more easily, the dominance that certain airlines have over other ones regarding total market occupancy.

To culminate this report I would like to provide insight regarding the share of the flights conducted by American Airlines month by month, to see if their total market participation has an increasing or decreasing trend. Ideally, this analysis is done year by year, but a year within year analysis can give us certain information on whether it is necessary to take action to gain more market occupation, or if we are going in the right tracks.

Firstly, we know from our “flights” database that there were 336,776 flights departing NYC in 2013. If we want to know how many were there in every month, using the “flights” data frame, and filtering the American Airlines flights, I can do it by applying filters to the .groupby function as follows:

```
airline_month_counts = flights[flights['carrier'] ==  
"AA"].groupby(['carrier', 'month'])['month'].count()
```

```
carrier  month  
AA       1      2794  
         2      2517  
         3      2787  
         4      2722  
         5      2803  
         6      2757  
         7      2882  
         8      2856  
         9      2614  
        10      2715  
        11      2577  
        12      2705  
Name: month, dtype: int64
```

To compare it to the rest of the flights, it is necessary to do the same without the filter, which gives us the following data:

```
total_month_counts = flights.groupby('month')['month'].count()  
  
total_month_counts  
month  
1      27004  
2      24951  
3      28834  
4      28330  
5      28796  
6      28243  
7      29425  
8      29327  
9      27574  
10     28889  
11     27268  
12     28135  
Name: month, dtype: int64
```

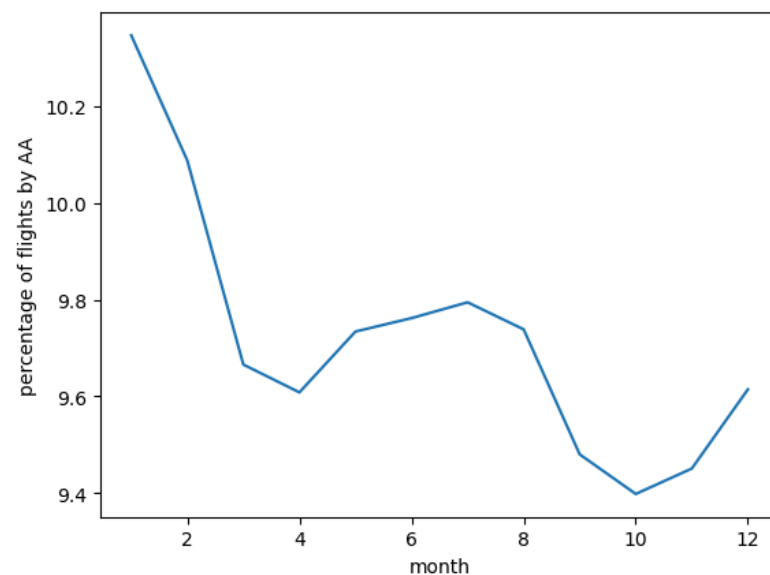
Finally, we apply a mathematical formula to obtain the percentage of participation each month

```
airline_month_percents = airline_month_counts /  
total_month_counts * 100
```

airline_month_percents		
carrier	month	
AA	1	10.346615
	2	10.087772
	3	9.665672
	4	9.608189
	5	9.733991
	6	9.761711
	7	9.794393
	8	9.738466
	9	9.479945
	10	9.398041
	11	9.450638
	12	9.614359
Name: month, dtype: float64		

and for better visualization, we rename the last column, turn it back into a dataframe and display it as a linear graph

```
ampf=airline_month_percents.to_frame()
ampf= ampf.rename(columns={0: 'percentage of flights by AA'})
sns.lineplot(data= ampf, x="month", y='percentage of flights by
AA')
```



This table turns out insightful when seeking strategies to increase market participation. For instance, American Airlines could significantly improve their market position for the months after february, since they started with over 10.2 percent. We also can assume that the actions being taken at the end of the year are the right ones, since the closing trend of the year is going upwards, growing from 9.4 in October to 9.6 in December.



## Ethical and communitarian commitment

To me, integrity represents the combination of values and morals that define each person. It can be different to each and every one of us, but the most important thing is to stand by it and respect it in others as well. To me personally, the phrase “integrity means to do the right thing even when no one is watching” comes to mind.

Making ethical use of data in business is an ongoing subject that becomes more relevant every day (Cote, 2021). The computational processes run every day faster, and other technological advancements allow us to have unimaginable amounts of information at our disposal at any time. Every company must decide on their own what ethical code will they base their data operations on (Grennan, Edquist, Rowshankish, 2022), but to me the ethical use that we give to data has to do with two key points, how we get it, and who we share it with (Abdullahi, 2022). To act with respect and honesty, I commit to always asking questions such as:

- Was the data obtained forcefully?
- Was the data stolen?
- Would this data cause damage if it were to be revealed?

And many others, that allow me to be at ease with the decision of accessing, manipulating, and visualizing data.

## References

- Abdullahi, A. (2022, August 2). *Why data ethics are important for your business*. IT Business Edge. Retrieved May 6, 2023, from <https://www.itbusinessedge.com/business-intelligence/data-ethics-framework/>
- Cothe, C. (2021, March 16). *5 principles of data ethics for business*. Business Insights Blog. Retrieved May 6, 2023, from <https://online.hbs.edu/blog/post/data-ethics>
- Edquist, A., Grennan, L., Griffiths, S., & Rowshankish, K. (2022, September 23). *Data ethics: What it means and what it takes*. McKinsey & Company. Retrieved May 6, 2023, from <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/data-ethics-what-it-means-and-what-it-takes>
- Google Colaboratory used:  
[https://colab.research.google.com/drive/1GzQX95ZwvjCBeAQhM3JZ\\_Mlarb\\_ICx9-?usp=sharing](https://colab.research.google.com/drive/1GzQX95ZwvjCBeAQhM3JZ_Mlarb_ICx9-?usp=sharing)