

# Exploració de les dades

## Descripció

- Familiaritza't amb les tècniques d'exploració de les dades mitjançant la estructura de dades, Dataframe amb la llibreria Pandas.

## NIVELL 1

### Exercici 1

Descarrega el data set Airlines Delay: Airline on-time statistics and delay causes i carrega'l a un pandas Dataframe. Explora les dades que conté, i queda't únicament amb les columnes que consideris rellevants.

```
In [1]: # Crido a les llibreries de pandas i matplotlib i numpy per poder treballar tranquilament
# Faig entrar l'arxiu CSV gràcies a pandas

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

delayedFlights = pd.read_csv(r'C:\Users\Anna\DataScience\SPRINTS\SPRINT 2\Sprint2_T05\DelayedFlights.csv')
delayedFlights[:]
```

```
Out[1]:
```

	Unnamed: 0	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier
0	0	2008	1	3	4	2003.0	1955	2211.0	2225	
1	1	2008	1	3	4	754.0	735	1002.0	1000	
2	2	2008	1	3	4	628.0	620	804.0	750	
3	4	2008	1	3	4	1829.0	1755	1959.0	1925	
4	5	2008	1	3	4	1940.0	1915	2121.0	2110	
...	...	...	...	...	...	...	...	...	...	...
1936753	7009710	2008	12	13	6	1250.0	1220	1617.0	1552	
1936754	7009717	2008	12	13	6	657.0	600	904.0	749	
1936755	7009718	2008	12	13	6	1007.0	847	1149.0	1010	
1936756	7009726	2008	12	13	6	1251.0	1240	1446.0	1437	
1936757	7009727	2008	12	13	6	1110.0	1103	1413.0	1418	

1936758 rows × 30 columns

```
In [2]: # Primer vull saber la mida de la taula (files, columnes)

print (delayedFlights.shape)
```

(1936758, 30)

```
In [3]: # Imprimeixo la funció info per saber de què està compost el dataframe
```

```
delayedFlights.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1936758 entries, 0 to 1936757
Data columns (total 30 columns):
#   Column                Dtype
---  -
0   Unnamed: 0            int64
1   Year                  int64
2   Month                 int64
3   DayofMonth            int64
4   DayOfWeek             int64
5   DepTime               float64
6   CRSDepTime            int64
7   ArrTime               float64
8   CRSArrTime            int64
9   UniqueCarrier         object
10  FlightNum             int64
11  TailNum               object
12  ActualElapsedTime     float64
13  CRSElapsedTime        float64
14  AirTime               float64
15  ArrDelay              float64
16  DepDelay              float64
17  Origin                object
18  Dest                  object
19  Distance              int64
20  TaxiIn                float64
21  TaxiOut               float64
22  Cancelled             int64
23  CancellationCode      object
24  Diverted              int64
25  CarrierDelay          float64
26  WeatherDelay          float64
27  NASDelay              float64
28  SecurityDelay          float64
29  LateAircraftDelay     float64
dtypes: float64(14), int64(11), object(5)
memory usage: 443.3+ MB
```

```
In [4]: # Aquesta funció ens ajuda a imprimir només els noms de les columnes per visualitzar si he
delayedFlights.columns
```

```
# El nom que canviaria aquí seria el primer, unnamed: 0 , però com que la vull eliminar me
```

```
Out[4]: Index(['Unnamed: 0', 'Year', 'Month', 'DayofMonth', 'DayOfWeek', 'DepTime',
            'CRSDepTime', 'ArrTime', 'CRSArrTime', 'UniqueCarrier', 'FlightNum',
            'TailNum', 'ActualElapsedTime', 'CRSElapsedTime', 'AirTime', 'ArrDelay',
            'DepDelay', 'Origin', 'Dest', 'Distance', 'TaxiIn', 'TaxiOut',
            'Cancelled', 'CancellationCode', 'Diverted', 'CarrierDelay',
            'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay'],
            dtype='object')
```

```
In [5]: #Ara vull saber si els valors d'alguna columna són TOTS Nuls, cosa que no m'aportarien cap
print(delayedFlights.isnull())
print("_____")
print(delayedFlights.count())

print ("Si hi ha un total de 1936758 files amb informació, les columnes que no coincideixi:
print("_____")
print(delayedFlights.isnull().sum())
```

```
print("A simple vista, podria eliminar algunas columnes que no em diuen res en quant als ")
```

	Unnamed: 0	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	\
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
1936753	False	False	False	False	False	False	False	
1936754	False	False	False	False	False	False	False	
1936755	False	False	False	False	False	False	False	
1936756	False	False	False	False	False	False	False	
1936757	False	False	False	False	False	False	False	

	ArrTime	CRSArrTime	UniqueCarrier	...	TaxiIn	TaxiOut	Cancelled	\
0	False	False	False	...	False	False	False	
1	False	False	False	...	False	False	False	
2	False	False	False	...	False	False	False	
3	False	False	False	...	False	False	False	
4	False	False	False	...	False	False	False	
...	...	...	...	...	...	...	...	
1936753	False	False	False	...	False	False	False	
1936754	False	False	False	...	False	False	False	
1936755	False	False	False	...	False	False	False	
1936756	False	False	False	...	False	False	False	
1936757	False	False	False	...	False	False	False	

	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	\
0	False	False	True	True	True	
1	False	False	True	True	True	
2	False	False	True	True	True	
3	False	False	False	False	False	
4	False	False	True	True	True	
...	...	...	...	...	...	
1936753	False	False	False	False	False	
1936754	False	False	False	False	False	
1936755	False	False	False	False	False	
1936756	False	False	True	True	True	
1936757	False	False	True	True	True	

	SecurityDelay	LateAircraftDelay
0	True	True
1	True	True
2	True	True
3	False	False
4	True	True
...	...	...
1936753	False	False
1936754	False	False
1936755	False	False
1936756	True	True
1936757	True	True

[1936758 rows x 30 columns]

---

Unnamed: 0	1936758
Year	1936758
Month	1936758
DayofMonth	1936758
DayOfWeek	1936758
DepTime	1936758
CRSDepTime	1936758
ArrTime	1929648

```

CRSArrTime      1936758
UniqueCarrier   1936758
FlightNum       1936758
TailNum        1936753
ActualElapsedTime 1928371
CRSElapsedTime  1936560
AirTime        1928371
ArrDelay       1928371
DepDelay       1936758
Origin         1936758
Dest          1936758
Distance       1936758
TaxiIn        1929648
TaxiOut        1936303
Cancelled      1936758
CancellationCode 1936758
Diverted       1936758
CarrierDelay   1247488
WeatherDelay   1247488
NASDelay       1247488
SecurityDelay  1247488
LateAircraftDelay 1247488

```

```
dtype: int64
```

Si hi ha un total de 1936758 files amb informació, les columnes que no coincideixin amb aquest número vol dir que tenen valors nuls, les quals les hauria d'estudiar més d'aprop d'e per què són nuls

---

```

Unnamed: 0      0
Year            0
Month           0
DayofMonth      0
DayOfWeek       0
DepTime         0
CRSDepTime      0
ArrTime         7110
CRSArrTime      0
UniqueCarrier   0
FlightNum       0
TailNum         5
ActualElapsedTime 8387
CRSElapsedTime  198
AirTime         8387
ArrDelay        8387
DepDelay        0
Origin          0
Dest            0
Distance        0
TaxiIn          7110
TaxiOut         455
Cancelled       0
CancellationCode 0
Diverted        0
CarrierDelay    689270
WeatherDelay    689270
NASDelay        689270
SecurityDelay   689270
LateAircraftDelay 689270

```

```
dtype: int64
```

A simple vista, podria eliminar algunes columnes que no em diuen res en quant als retards dels vols, com per exemple l'any, ja que tots els vols són del 2008, la columna Unnamed, ja que tinc el número de vol que em pot fer la mateixa funció, etc.

```
In [6]: # Elimino les columnes que crec que no m'aporten molta informació o que no en puc extreure
```

```

delayedFlights = delayedFlights.drop(['Unnamed: 0', 'Year', 'Cancelled', 'CancellationCode'])
print(delayedFlights)

```

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	\
0	1	3	4	2003.0	1955	2211.0	
1	1	3	4	754.0	735	1002.0	
2	1	3	4	628.0	620	804.0	
3	1	3	4	1829.0	1755	1959.0	
4	1	3	4	1940.0	1915	2121.0	
...	...	...	...	...	...	...	
1936753	12	13	6	1250.0	1220	1617.0	
1936754	12	13	6	657.0	600	904.0	
1936755	12	13	6	1007.0	847	1149.0	
1936756	12	13	6	1251.0	1240	1446.0	
1936757	12	13	6	1110.0	1103	1413.0	

	CRSArrTime	UniqueCarrier	FlightNum	ActualElapsedTime	...	\
0	2225	WN	335	128.0	...	
1	1000	WN	3231	128.0	...	
2	750	WN	448	96.0	...	
3	1925	WN	3920	90.0	...	
4	2110	WN	378	101.0	...	
...	...	...	...	...	...	
1936753	1552	DL	1621	147.0	...	
1936754	749	DL	1631	127.0	...	
1936755	1010	DL	1631	162.0	...	
1936756	1437	DL	1639	115.0	...	
1936757	1418	DL	1641	123.0	...	

	DepDelay	Origin	Dest	Distance	Diverted	CarrierDelay	WeatherDelay	\
0	8.0	IAD	TPA	810	0	NaN	NaN	
1	19.0	IAD	TPA	810	0	NaN	NaN	
2	8.0	IND	BWI	515	0	NaN	NaN	
3	34.0	IND	BWI	515	0	2.0	0.0	
4	25.0	IND	JAX	688	0	NaN	NaN	
...	...	...	...	...	...	...	...	
1936753	30.0	MSP	ATL	906	0	3.0	0.0	
1936754	57.0	RIC	ATL	481	0	0.0	57.0	
1936755	80.0	ATL	IAH	689	0	1.0	0.0	
1936756	11.0	IAD	ATL	533	0	NaN	NaN	
1936757	7.0	SAT	ATL	874	0	NaN	NaN	

	NASDelay	SecurityDelay	LateAircraftDelay
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	0.0	0.0	32.0
4	NaN	NaN	NaN
...	...	...	...
1936753	0.0	0.0	22.0
1936754	18.0	0.0	0.0
1936755	19.0	0.0	79.0
1936756	NaN	NaN	NaN
1936757	NaN	NaN	NaN

[1936758 rows x 23 columns]

```

In [7]: # He eliminat 7 columnes que crec que no les necessitaré.

# En els següents punts, faig alguns plots per practicar i veure una mica com va funcionar

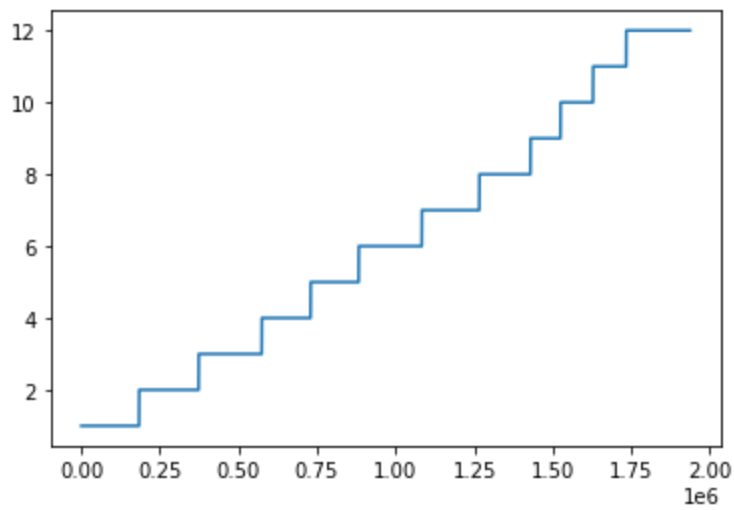
```

```

In [8]: delayedFlights['Month'].plot()

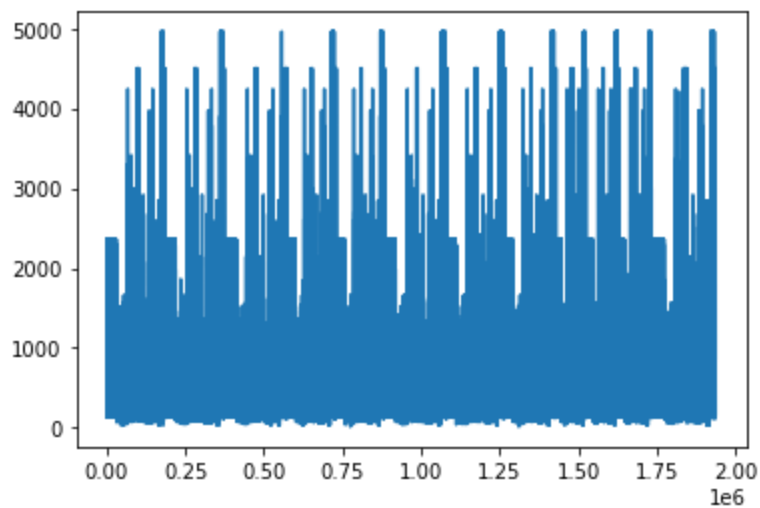
```

Out[8]: <AxesSubplot:>



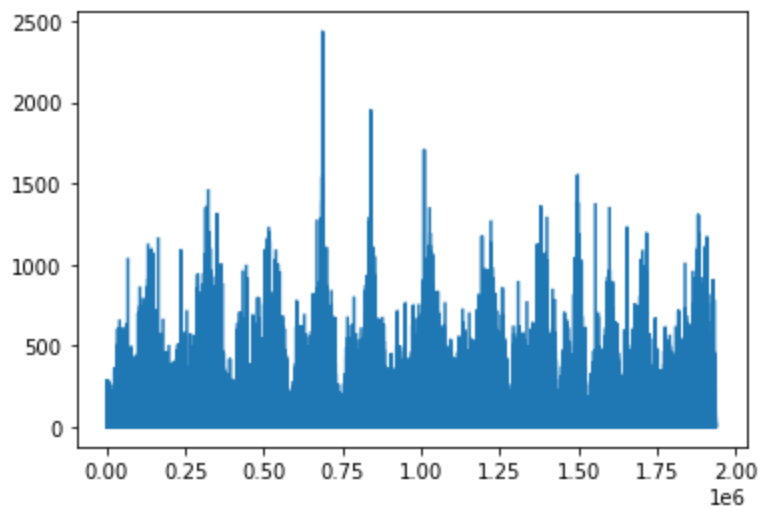
In [9]: `delayedFlights['Distance'].plot()`

Out[9]: <AxesSubplot:>



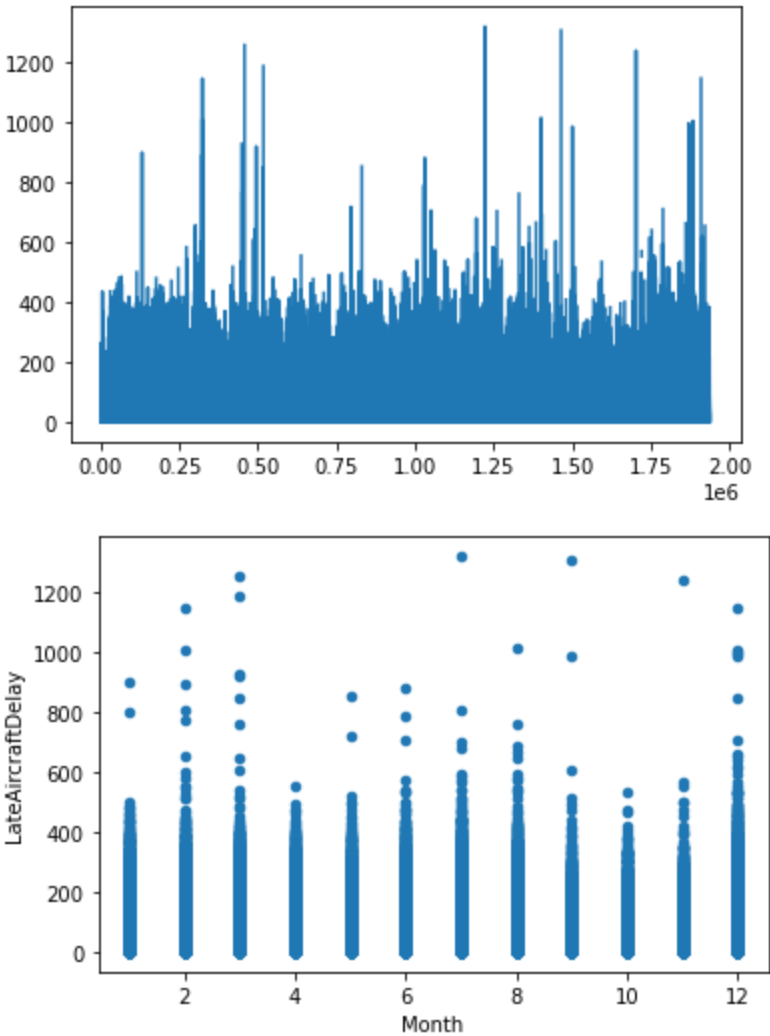
In [10]: `delayedFlights['CarrierDelay'].plot()`

Out[10]: <AxesSubplot:>



In [11]: `delayedFlights['LateAircraftDelay'].plot()  
delayedFlights.plot(kind = "scatter", y = "LateAircraftDelay", x = "Month")`

```
Out[11]: <AxesSubplot:xlabel='Month', ylabel='LateAircraftDelay'>
```



## Exercici 2

Fes un informe complet del data set:

- Resumeix estadísticament les columnes d'interès
- Troba quantes dades faltants hi ha per columna
- Crea columnes noves (velocitat mitjana del vol, si ha arribat tard o no...)
- Taula de les aerolínies amb més endarreriments acumulats
- Quins són els vols més llargs? I els més endarrerits?
- Etc.

```
In [12]: # Imprimeixo les dades bàsiques que em fa la funció describe()

delayedFlights.describe()
```

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	Fligh
count	1.936758e+06	1.936758e+06	1.936758e+06	1.936758e+06	1.936758e+06	1.929648e+06	1.936758e+06	1.936758e+06
mean	6.111106e+00	1.575347e+01	3.984827e+00	1.518534e+03	1.467473e+03	1.610141e+03	1.634225e+03	2.18426e+03
std	3.482546e+00	8.776272e+00	1.995966e+00	4.504853e+02	4.247668e+02	5.481781e+02	4.646347e+02	1.94470e+03
min	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	Fligh
25%	3.000000e+00	8.000000e+00	2.000000e+00	1.203000e+03	1.135000e+03	1.316000e+03	1.325000e+03	6.10000
50%	6.000000e+00	1.600000e+01	4.000000e+00	1.545000e+03	1.510000e+03	1.715000e+03	1.705000e+03	1.54300
75%	9.000000e+00	2.300000e+01	6.000000e+00	1.900000e+03	1.815000e+03	2.030000e+03	2.014000e+03	3.42200
max	1.200000e+01	3.100000e+01	7.000000e+00	2.400000e+03	2.359000e+03	2.400000e+03	2.400000e+03	9.74200

```
In [17]: # Imprimeixo la correlació que em fa la funció corr()
delayedFlights1 = delayedFlights.dropna()
delayedFlights1.corr
```

Out[17]:	<bound method DataFrame.corr of							Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime
	ArrTime \											
	3	1	3	4	1829.0	1755	1959.0					
	5	1	3	4	1937.0	1830	2037.0					
	7	1	3	4	1644.0	1510	1845.0					
	9	1	3	4	1452.0	1425	1640.0					
	11	1	3	4	1323.0	1255	1526.0					
	...	...	...	...	...	...	...					
	1936751	12	13	6	921.0	830	1112.0					
	1936752	12	13	6	1552.0	1520	1735.0					
	1936753	12	13	6	1250.0	1220	1617.0					
	1936754	12	13	6	657.0	600	904.0					
	1936755	12	13	6	1007.0	847	1149.0					
	CRSArrTime UniqueCarrier FlightNum ActualElapsedTime ... \											
	3	1925	WN	3920	90.0	...						
	5	1940	WN	509	240.0	...						
	7	1725	WN	1333	121.0	...						
	9	1625	WN	675	228.0	...						
	11	1510	WN	4	123.0	...						
	...	...	...	...	...	...						
	1936751	1008	DL	1616	111.0	...						
	1936752	1718	DL	1620	43.0	...						
	1936753	1552	DL	1621	147.0	...						
	1936754	749	DL	1631	127.0	...						
	1936755	1010	DL	1631	162.0	...						
	DepDelay Origin Dest Distance Diverted CarrierDelay WeatherDelay \											
	3	34.0	IND	BWI	515	0	2.0	0.0				
	5	67.0	IND	LAS	1591	0	10.0	0.0				
	7	94.0	IND	MCO	828	0	8.0	0.0				
	9	27.0	IND	PHX	1489	0	3.0	0.0				
	11	28.0	IND	TPA	838	0	0.0	0.0				
	...	...	...	...	...	...	...	...				
	1936751	51.0	ATL	PBI	545	0	51.0	0.0				
	1936752	32.0	HSV	ATL	151	0	0.0	0.0				
	1936753	30.0	MSP	ATL	906	0	3.0	0.0				
	1936754	57.0	RIC	ATL	481	0	0.0	57.0				
	1936755	80.0	ATL	IAH	689	0	1.0	0.0				
	NASDelay SecurityDelay LateAircraftDelay											
	3	0.0	0.0	32.0								
	5	0.0	0.0	47.0								
	7	0.0	0.0	72.0								
	9	0.0	0.0	12.0								
	11	0.0	0.0	16.0								
	...	...	...	...								
	1936751	13.0	0.0	0.0								
	1936752	0.0	0.0	17.0								
	1936753	0.0	0.0	22.0								



```

1936754      18.0      0.0      0.0
1936755      19.0      0.0      79.0

```

```
[1247488 rows x 23 columns]>
```

In [14]:

```

'''
Resto hora prevista d'arribada amb hora d'arribada real per veure si hi ha
hagut retard o no, si el resultat és negatiu, vol dir que no hi ha hagut
retard, i si és positiu és que sí que hi ha hagut retard
'''

retard = np.subtract(delayedFlights.ArrTime , delayedFlights.CRSArrTime)
print(retard)

print("El retard més gran va ser de", retard.max(), "minuts, és a dir, de", retard.max()/60)
# No sé si és possible un retard tant gran o estic fent alguna cosa malament, potser per a

3          34.0
5          97.0
7         120.0
9          15.0
11         16.0
...
1936751    104.0
1936752     17.0
1936753     65.0
1936754    155.0
1936755    139.0
Length: 1247488, dtype: float64
El retard més gran va ser de 1844.0 minuts, és a dir, de 30.733333333333334 h.

```

In [15]:

```

# Inserto el resultat "TotalDelay" al dataframe principal

delayedFlights1 = delayedFlights.assign(totalTimeDelayed = retard).values)

File "C:\Users\Anna\AppData\Local\Temp\ipykernel_15244\3379182732.py", line 3
    delayedFlights1 = delayedFlights.assign(totalTimeDelayed = retard).values)
                                                                    ^
SyntaxError: unmatched ')'

```

In [ ]:

```

'''
Faig un true/false per després fer un contador en bucle per saber quants vols
hi ha hagut en retard
'''

condicio = (retard[:] > 0)

print(condicio)

```

In [ ]:

```

'''
contador = retard.apply(
    lambda x: True if x[1] > 0 else False)
numeroRetards = len(contador[contador == True].index)

print(numeroRetards)

contador = 0

for x in range(len(retard)):
    if retard.iloc[1,:]:
        contador +=1

```

```
print(contador)

index, counts = np.unique(
    retard.to_numpy(),
    return_counts=True
)
suma = retard.Series(counts, index)
'''
```

In [ ]:

## Exercici 3

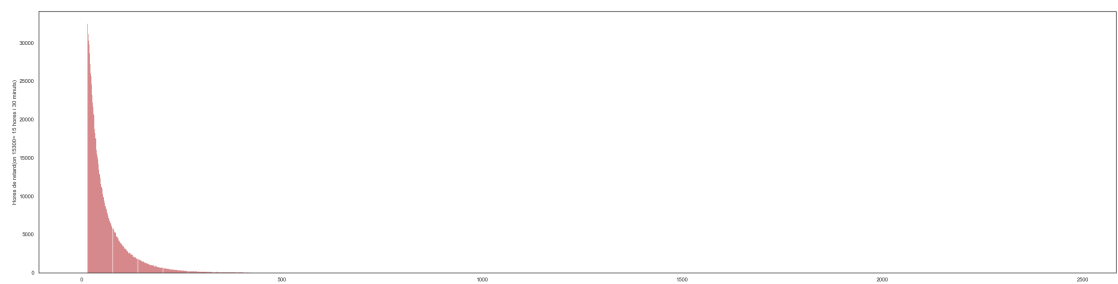
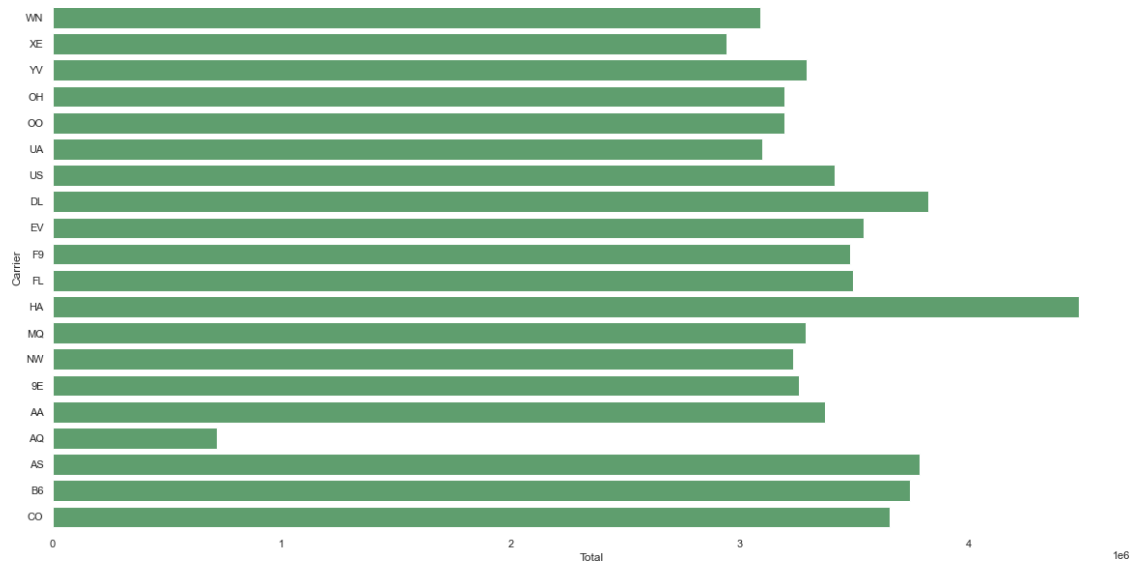
Exporta el data set net i amb les noves columnes a Excel.

In [ ]:

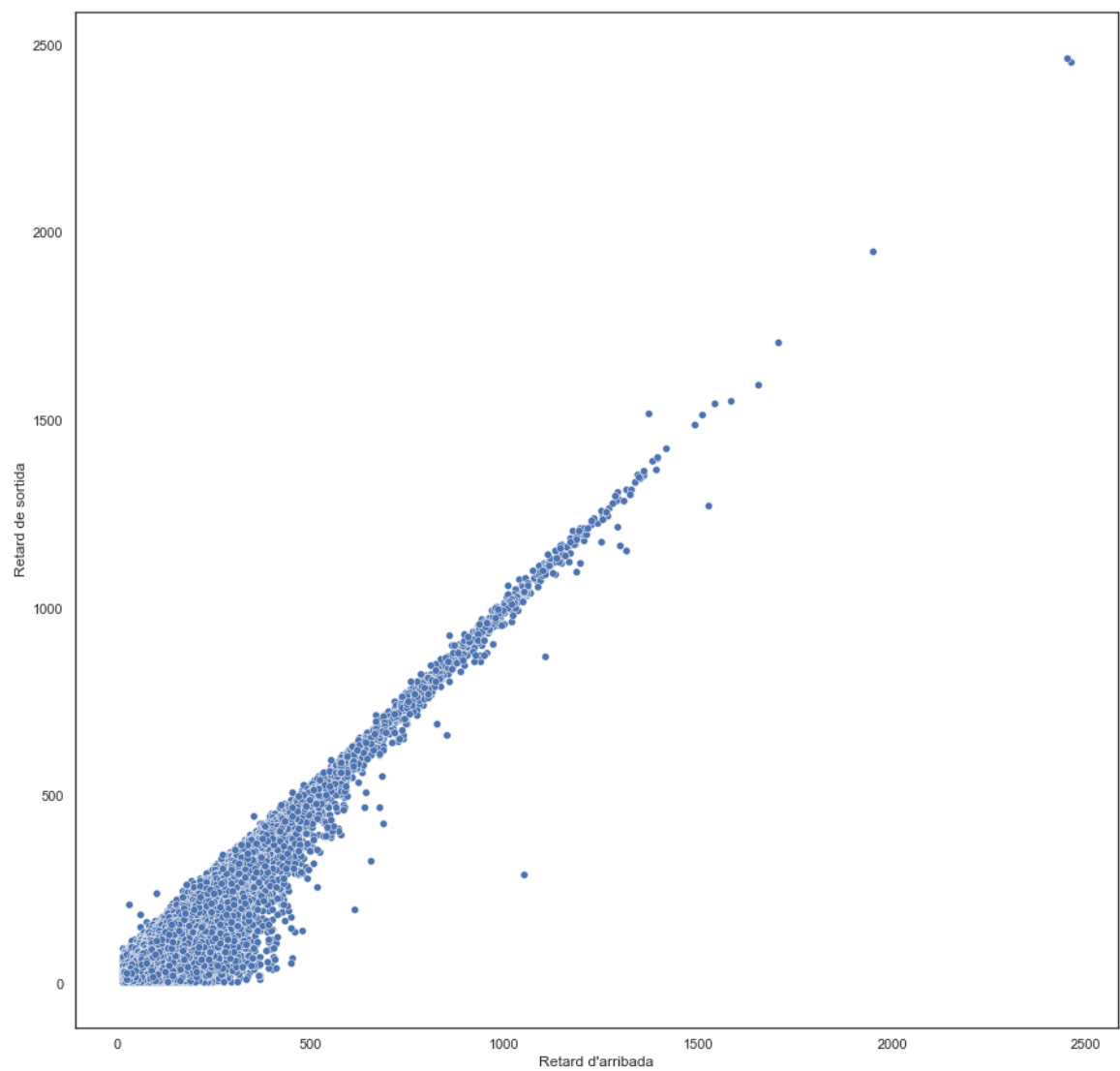
```
'''
# determining the name of the file
docExcel = 'TotalDelayedTime.xlsx'

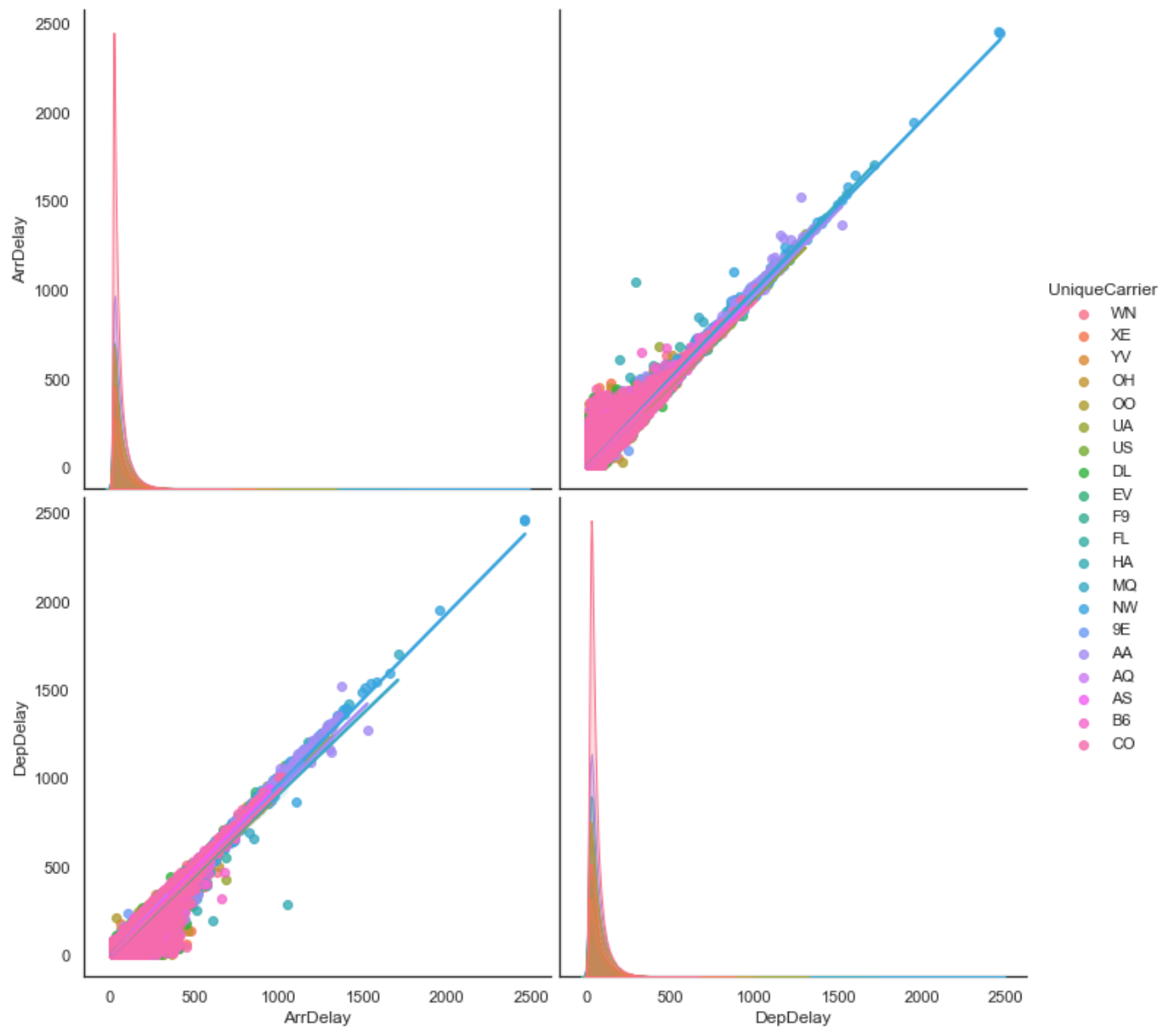
# saving the excel
delayedFlights.to_excel(docExcel)
'''
```

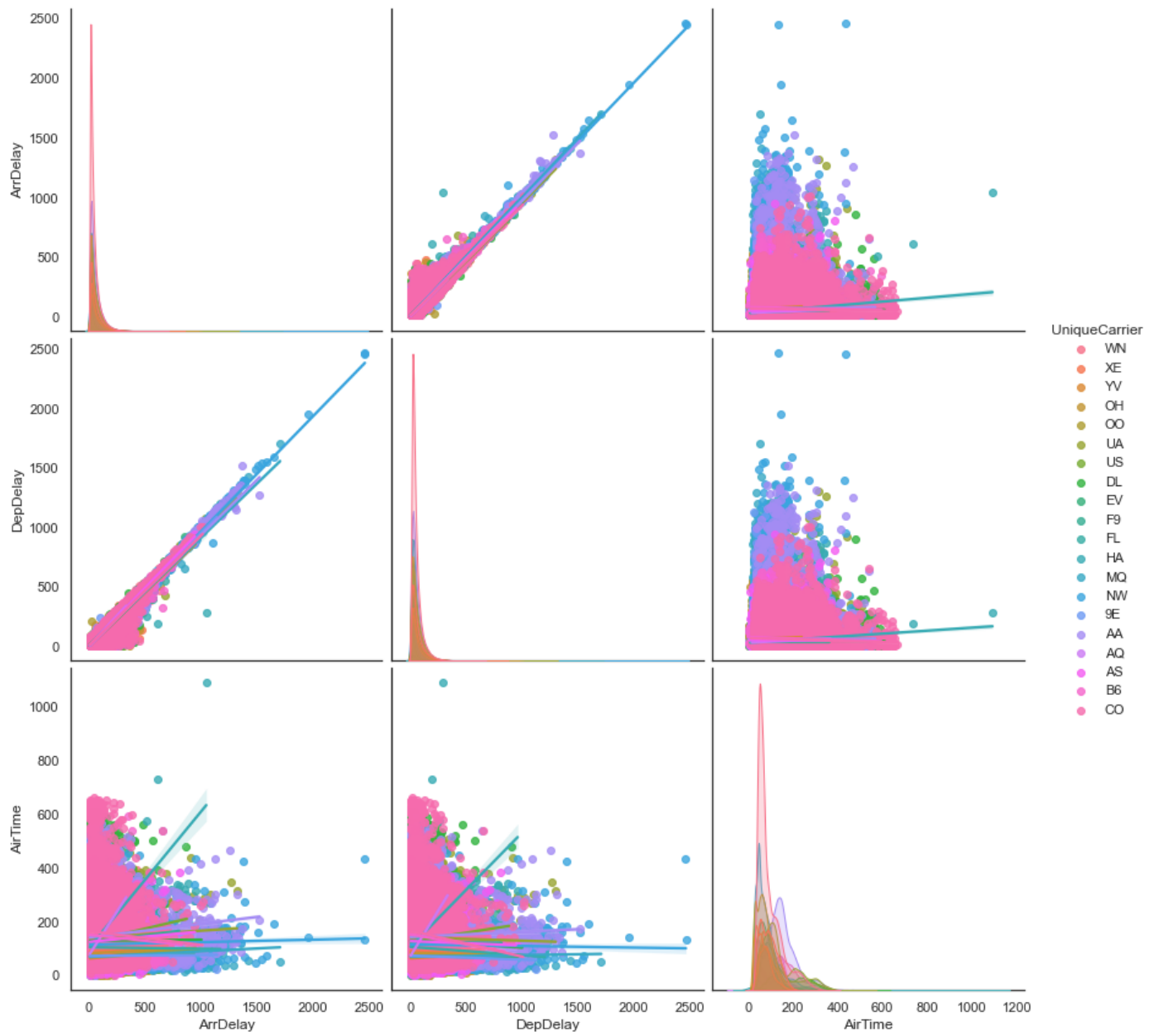
## IMATGES STRING3











In [ ]: