

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 [10]: # 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[10]:
```

| | 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 [11]: # Primer vull saber la mida de la taula (files, columnes)

print (delayedFlights.shape)
```

(1936758, 30)

```
In [12]: # 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 [13]: # 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[13]: 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 [14]: #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 [15]: `# 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 [25]: # 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

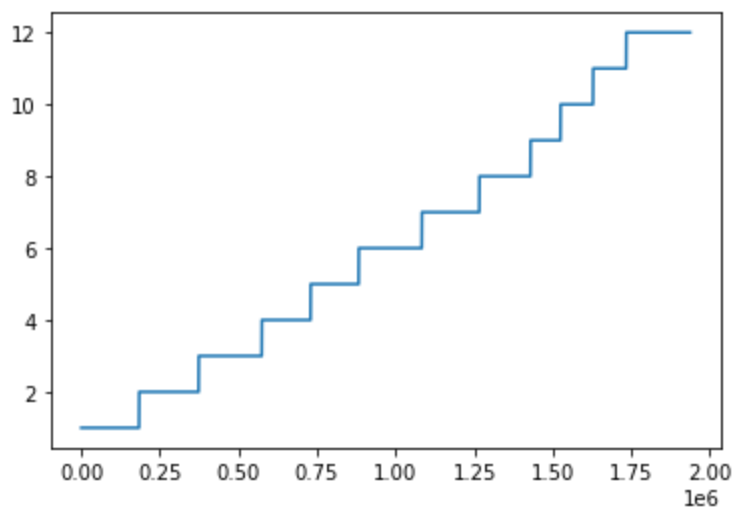
```

```

In [26]: delayedFlights['Month'].plot()

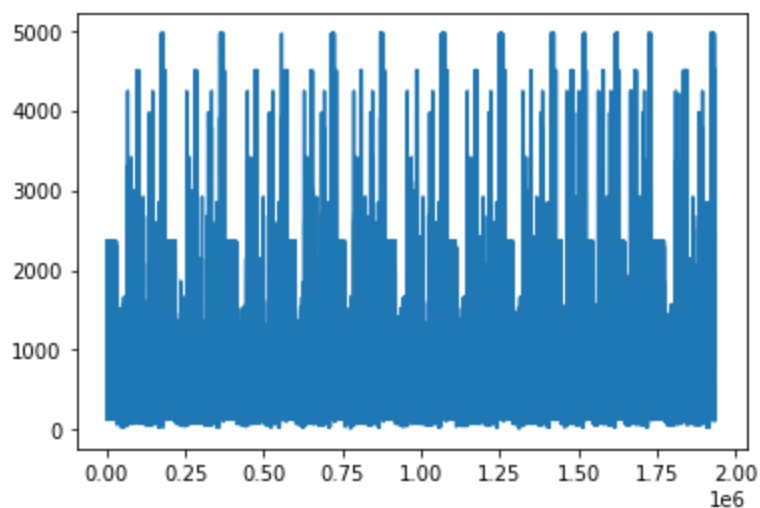
```

Out[26]: <AxesSubplot:>



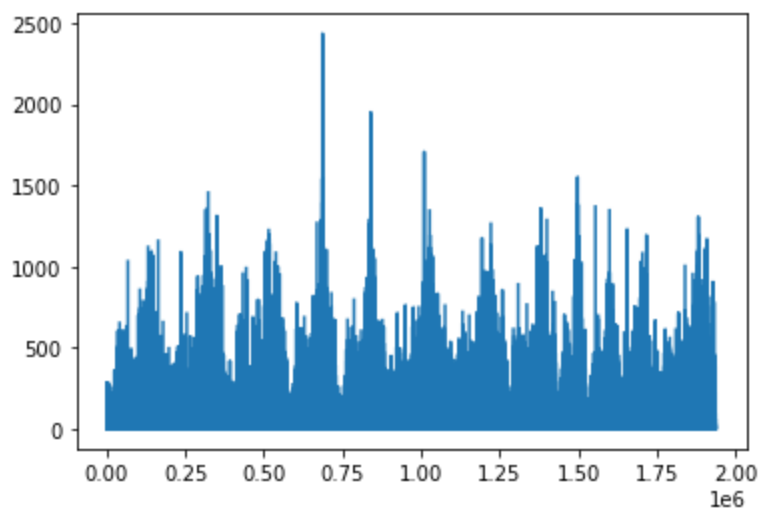
In [27]: `delayedFlights['Distance'].plot()`

Out[27]: <AxesSubplot:>



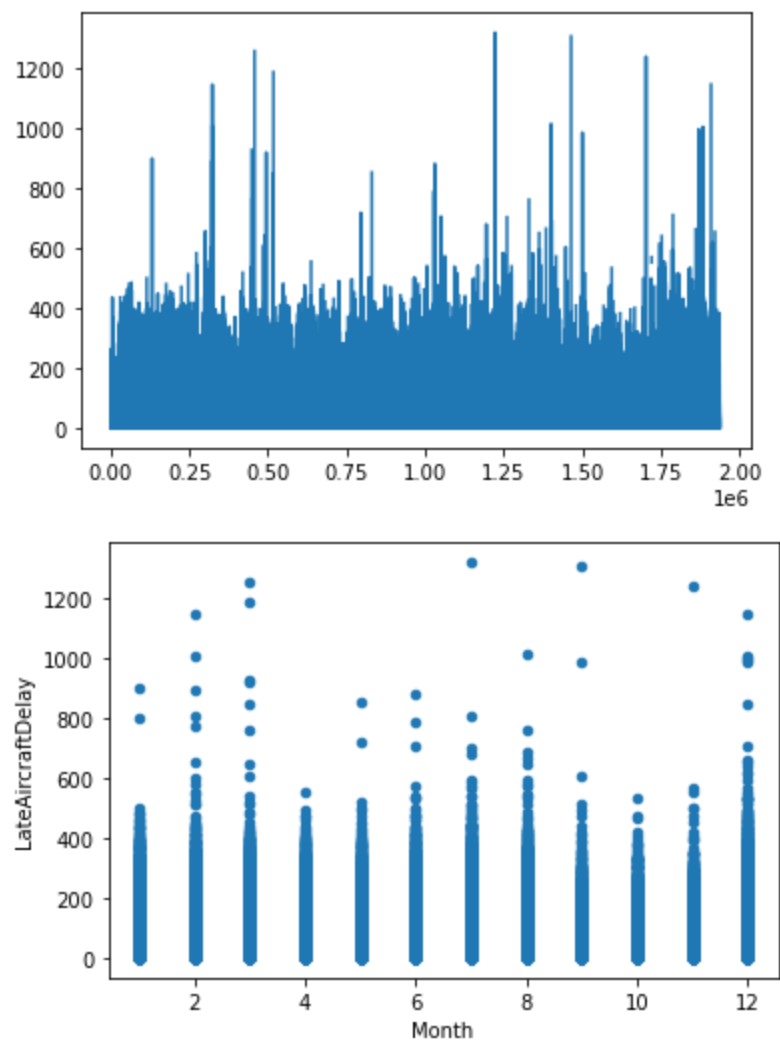
In [28]: `delayedFlights['CarrierDelay'].plot()`

Out[28]: <AxesSubplot:>



In [102...]: `delayedFlights['LateAircraftDelay'].plot()
delayedFlights.plot(kind = "scatter", y = "LateAircraftDelay", x = "Month")`

Out[102...<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 [57]: # Imprimeixo les dades bàsiques que em fa la funció describe()
delayedFlights.describe()
```

Out[57]:

| | 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 | 1.93675 |
| mean | 6.111106e+00 | 1.575347e+01 | 3.984827e+00 | 1.518534e+03 | 1.467473e+03 | 1.610141e+03 | 1.634225e+03 | 2.18426 |
| std | 3.482546e+00 | 8.776272e+00 | 1.995966e+00 | 4.504853e+02 | 4.247668e+02 | 5.481781e+02 | 4.646347e+02 | 1.94470 |
| min | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 0.000000e+00 | 1.000000e+00 | 0.000000e+00 | 1.00000 |

| | 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 [58]: # Imprimeixo la correlació que em fa la funció corr()

delayedFlights.corr()
```

Out[58]: **limit_output extension: Maximum message size of 10000 exceeded with 11487 characters**

```
In [84]: '''
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, "h.")
# No sé si és possible un retard tant gran o estic fent alguna cosa malament, pots
er per alguna cosa climàtica que es tanquéssin aeroports
```

```
0      -14.0
1         2.0
2        54.0
3        34.0
4        11.0
```

```
...
1936753    65.0
1936754   155.0
1936755   139.0
1936756     9.0
1936757    -5.0
```

Length: 1936758, dtype: float64

El retard més gran va ser de 2399.0 minuts, és a dir, de 39.983333333333334 h.

```
In [98]: # Inserto el resultat "TotalDelay" al dataframe principal

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

ValueError

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_18968\4276466275.py in <module>

2

3 #df1.assign(e=pd.Series(np.random.randn(sLength)).values)

```
----> 4 delayedFlights1 = delayedFlights.assign(totalTimeDelayed = pd.Series(np.subtrac
t(totalDelay = [delayedFlights.ArrTime , delayedFlights.CRSArrTime])).values)
```

ValueError: invalid number of arguments

```
In [85]: '''
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)
```

```
0      False
1      True
2      True
3      True
4      True

...
1936753    True
1936754    True
1936755    True
1936756    True
1936757    False
Length: 1936758, dtype: bool
```

```
In [88]: '''
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)
'''
```

```
-----
KeyError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_18968\1283071686.py in <module>
    22 suma = retard.Series(counts, index)
    23 '''
--> 24 suma= retard["True"].sum(axis=1)

~\anaconda3\lib\site-packages\pandas\core\series.py in __getitem__(self, key)
    940
    941         elif key_is_scalar:
--> 942             return self._get_value(key)
    943
    944         if is_hashable(key):

~\anaconda3\lib\site-packages\pandas\core\series.py in _get_value(self, label, takeabl
e)
    1049
```

```

1050 # Similar to Index.get_value, but we do not fall back to positional
-> 1051     loc = self.index.get_loc(label)
1052     return self.index._get_values_for_loc(self, loc, label)
1053

~\anaconda3\lib\site-packages\pandas\core\indexes\range.py in get_loc(self, key, metho
d, tolerance)
386         except ValueError as err:
387             raise KeyError(key) from err
--> 388         raise KeyError(key)
389     return super().get_loc(key, method=method, tolerance=tolerance)
390

```

KeyError: 'True'

In []:

Exercici 3

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

```

In [101... # determining the name of the file
docExcel = 'TotalDelayedTime.xlsx'

# saving the excel
delayedFlights.to_excel(docExcel)

```

```

-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_18968\3528203272.py in <module>
3
4 # saving the excel
----> 5 delayedFlights.to_excel(docExcel)

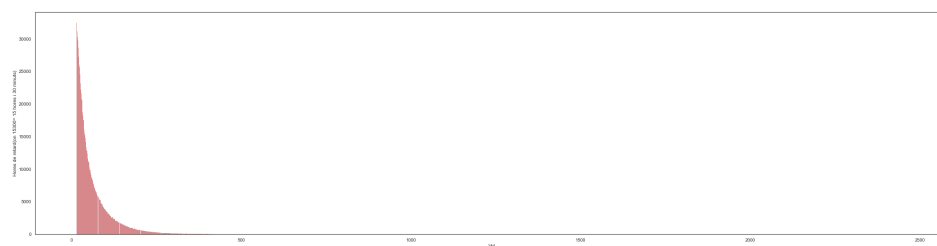
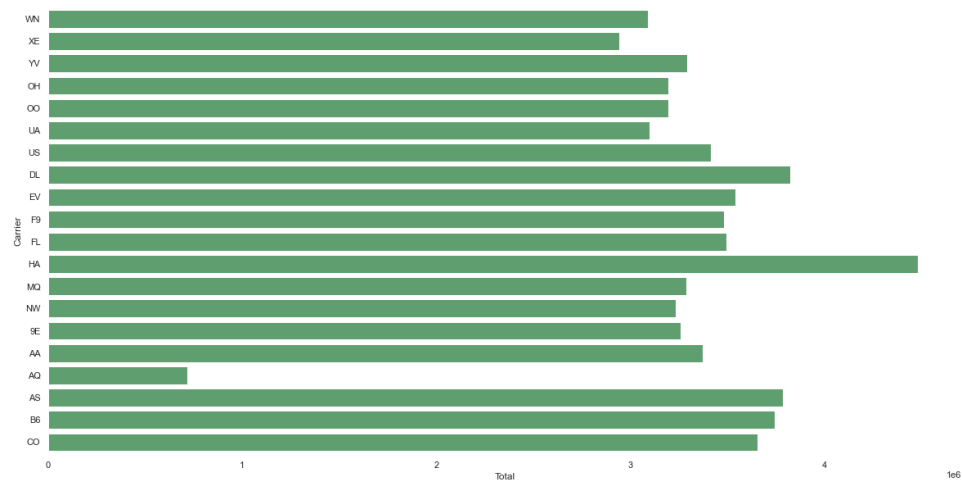
~\anaconda3\lib\site-packages\pandas\core\generic.py in to_excel(self, excel_writer, sh
eet_name, na_rep, float_format, columns, header, index, index_label, startrow, startco
l, engine, merge_cells, encoding, inf_rep, verbose, freeze_panes, storage_options)
2282         inf_rep=inf_rep,
2283     )
-> 2284     formatter.write(
2285         excel_writer,
2286         sheet_name=sheet_name,

~\anaconda3\lib\site-packages\pandas\io\formats\excel.py in write(self, writer, sheet_n
ame, startrow, startcol, freeze_panes, engine, storage_options)
821         num_rows, num_cols = self.df.shape
822         if num_rows > self.max_rows or num_cols > self.max_cols:
--> 823             raise ValueError(
824                 f"This sheet is too large! Your sheet size is: {num_rows}, {num
_cols} "
825                 f"Max sheet size is: {self.max_rows}, {self.max_cols}"

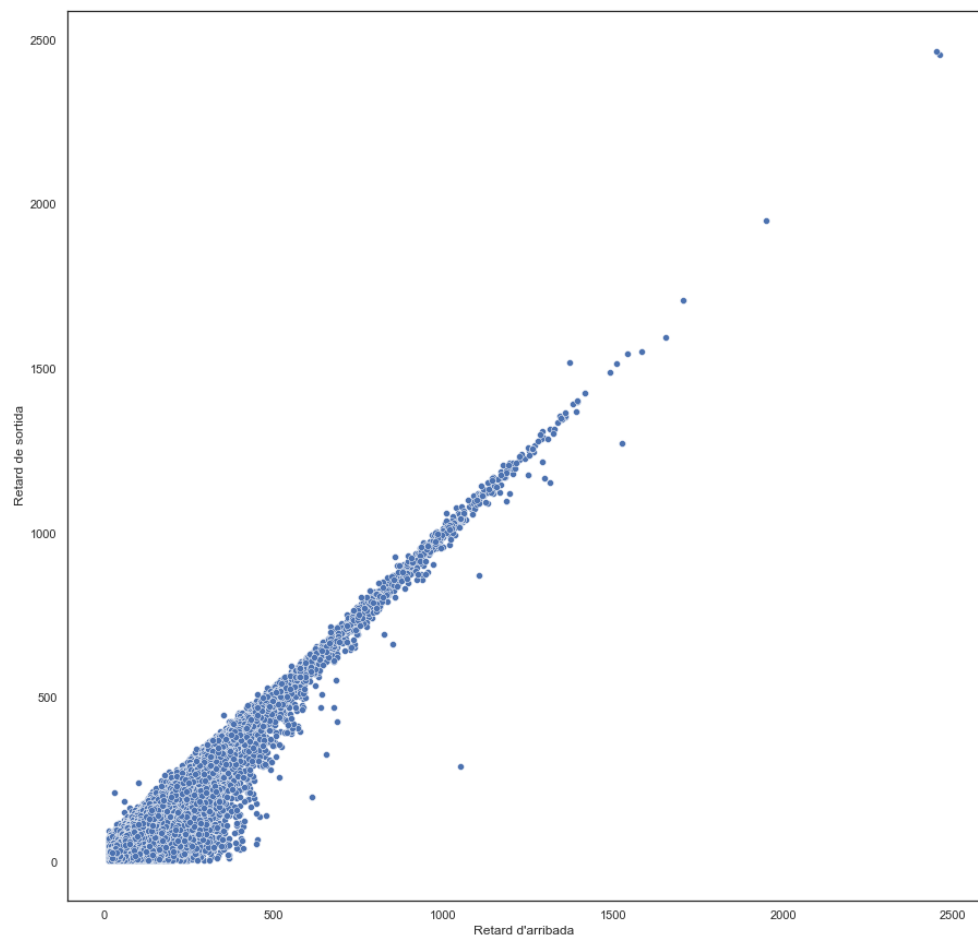
```

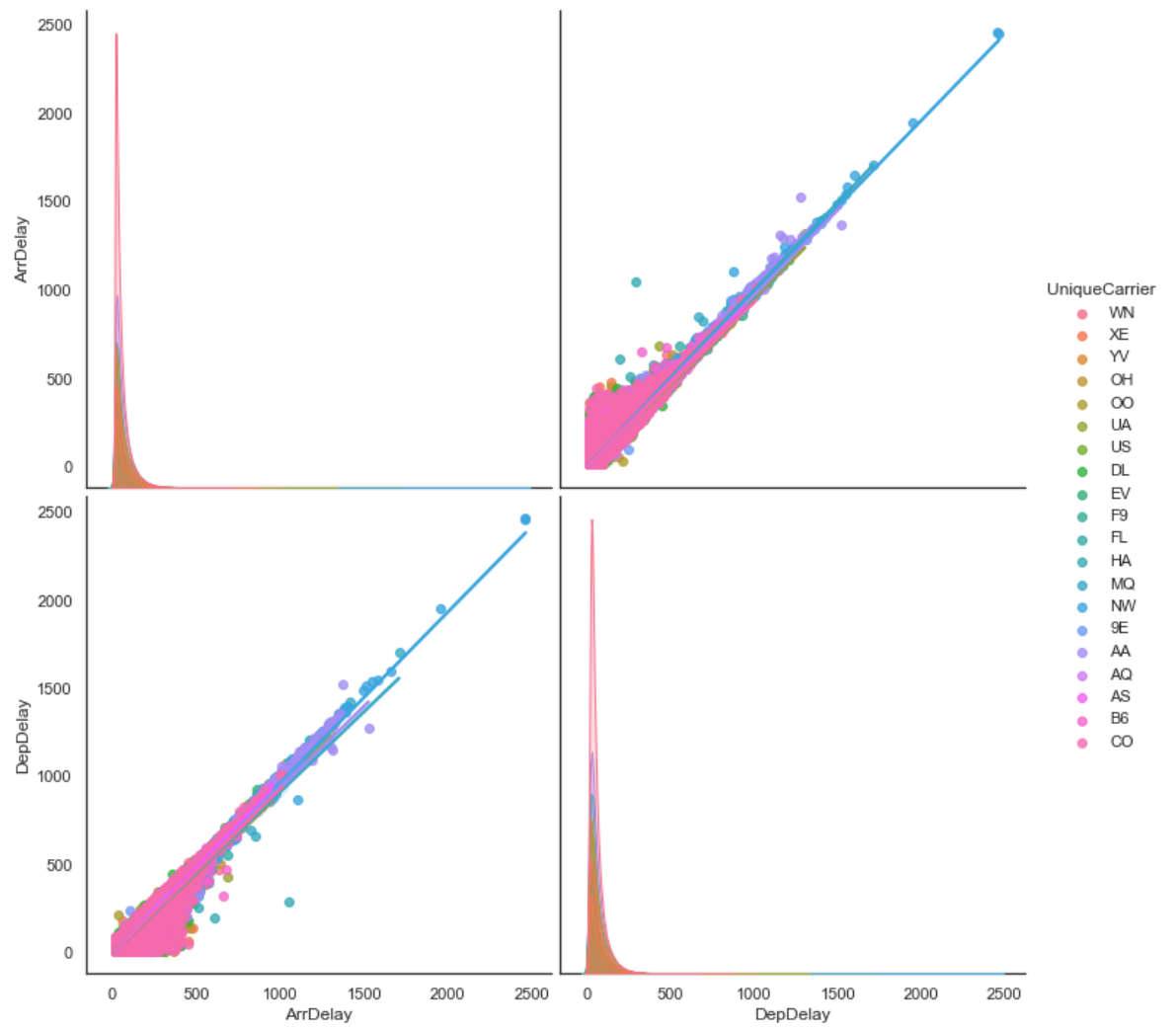
ValueError: This sheet is too large! Your sheet size is: 1936758, 23 Max sheet size is: 1048576, 16384

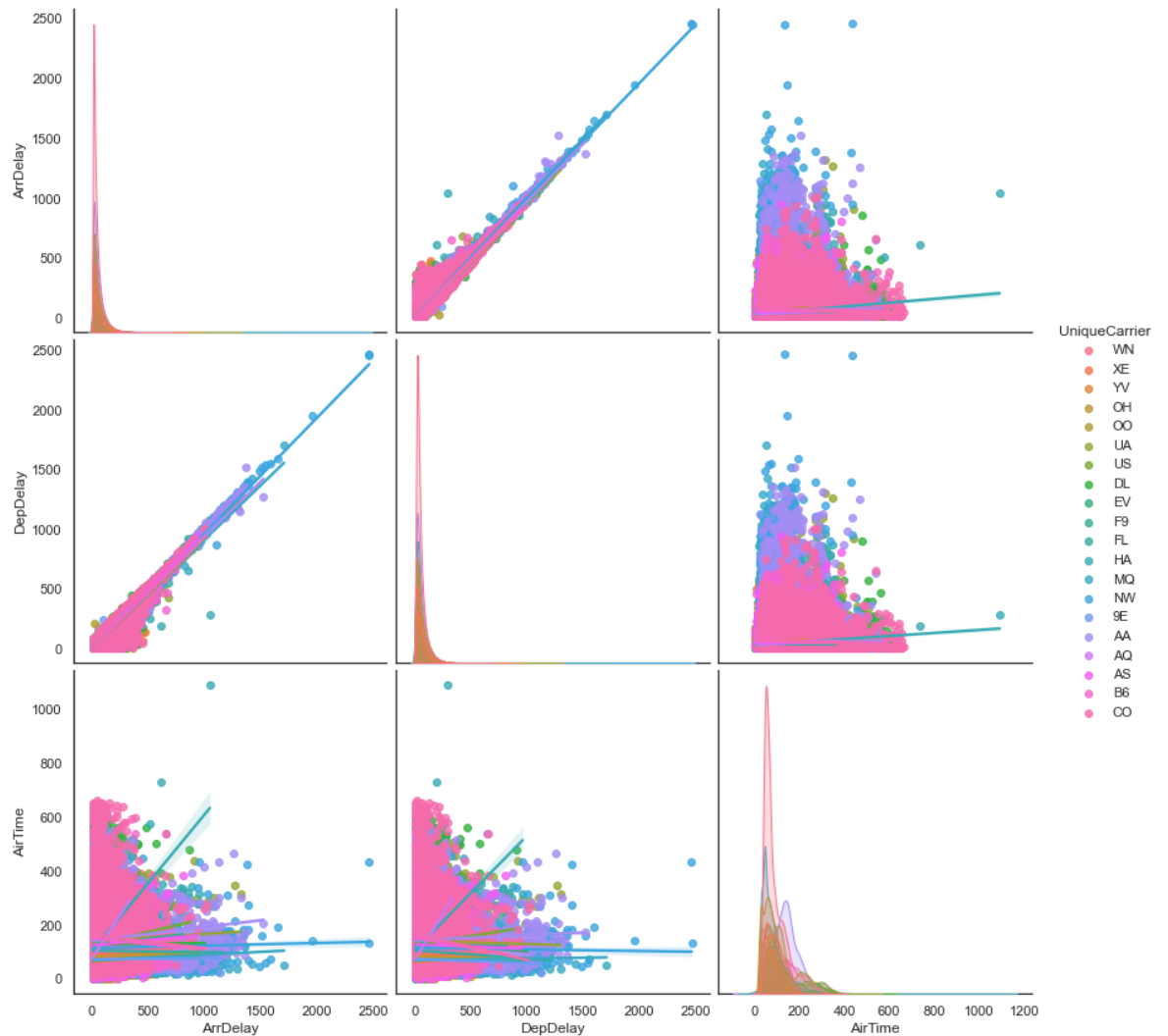
IMATGES STRING3











In []:

| | Month | DayofMonth | DayOfWeek | DepTime | CRSDepTime | ArrTime | CRSArrTime | FlightNum |
|--------------------------|-----------|------------|-----------|-----------|------------|-----------|------------|-----------|
| Month | 1.000000 | 0.059651 | 0.000088 | -0.007809 | -0.011367 | 0.001014 | 0.001367 | -0.000188 |
| DayofMonth | 0.059651 | 1.000000 | 0.017476 | 0.001014 | 0.001019 | 0.000833 | 0.000758 | -0.005912 |
| DayOfWeek | 0.000088 | 0.017476 | 1.000000 | 0.021924 | 0.027039 | 0.010913 | 0.017750 | -0.009769 |
| DepTime | -0.007809 | 0.001014 | 0.021924 | 1.000000 | 0.881598 | 0.458934 | 0.711513 | -0.024786 |
| CRSDepTime | -0.011367 | 0.001019 | 0.027039 | 0.881598 | 1.000000 | 0.396724 | 0.710303 | -0.054808 |
| ArrTime | 0.001014 | 0.000833 | 0.010913 | 0.458934 | 0.396724 | 1.000000 | 0.619385 | -0.013665 |
| CRSArrTime | 0.001367 | 0.000758 | 0.017750 | 0.711513 | 0.710303 | 0.619385 | 1.000000 | -0.060006 |
| FlightNum | -0.000188 | -0.005912 | -0.009769 | -0.024786 | -0.054808 | -0.013665 | -0.060006 | 1.000000 |
| ActualElapsedTime | 0.002684 | -0.000880 | 0.003072 | -0.047040 | -0.034925 | -0.013595 | 0.033203 | -0.322283 |
| CRSElapsedTime | 0.007046 | -0.000028 | 0.004954 | -0.044619 | -0.026388 | -0.012911 | 0.040117 | -0.335956 |
| AirTime | 0.000860 | -0.000244 | 0.004738 | -0.054831 | -0.036582 | -0.017684 | 0.025907 | -0.341250 |
| ArrDelay | -0.000897 | 0.004129 | 0.006123 | 0.127017 | 0.044447 | -0.050948 | 0.043078 | 0.061266 |
| DepDelay | 0.004769 | 0.005289 | 0.008538 | 0.139254 | 0.058875 | -0.053024 | 0.053706 | 0.051852 |
| Distance | 0.005498 | 0.000117 | 0.008138 | -0.056003 | -0.029517 | -0.027751 | 0.024335 | -0.356770 |
| Diverted | 0.006467 | 0.001190 | -0.001361 | -0.004632 | -0.009096 | -0.007204 | 0.000907 | -0.002885 |

| | Month | DayofMonth | DayOfWeek | DepTime | CRSDepTime | ArrTime | CRSArrTime | FlightNum |
|--------------|----------|------------|-----------|-----------|------------|-----------|------------|-----------|
| CarrierDelay | 0.000420 | -0.000947 | 0.010215 | -0.051948 | -0.107337 | -0.083981 | -0.096156 | 0.055712 |
| WeatherDelay | 0.006611 | 0.000916 | 0.005654 | 0.005307 | -0.009338 | -0.029860 | -0.005033 | 0.067488 |
| NASDelay | 0.011441 | 0.005644 | -0.006628 | 0.022530 | | | | |