

DATA SCIENCE

Sprint 3 : Programació numèrica, dataframes i anàlisi estadístic

Tasca M3 T02 - Exercicis amb Dataframes

- Realitzar anàlisis exploratòria de les dades
- Introducció a la llibreria Pandas, als Dataframes i els mètodes

Lliurament: Enviar l'URL a un repositori anomenat estructures\_Dataframe que contingui la solució.

EXERCICI 1

- 1.1 Descarrega el data set Airlines Delay: Airline on-time statistics and delay causes. Carrega'l a un Pandas Dataframe i explora les dades que conté.
- 1.2 Explica breument quines variables hi ha.
- 1.3 Queda't únicament amb les columnes que consideris rellevants, justificant la teva elecció.
- 1.4 Redueix la dimensió del dataset de manera aleatòria **per tal d'obtenir un dataset de només 200.000 registres. Tots els exercicis s'han de fer amb aquest dataset reduït.**

```
In [2]: import pandas as pd
import numpy as np
from scipy import stats
```

1.1 Descarrega el data set Airlines Delay: Airline on-time statistics and delay causes.

El descarreguem a un Pandas Dataframe i explorem les dades que conté :

```
In [3]: # Descarregant el Dataset en un Pandas Dataframe desde La ruta Local :
data = pd.read_csv('C:/Users/Buba/Documents/CURSOS-PROGRAMACION/IT-Academy/IT-DATA-SCIENCE/DS_Sprint3-Pandas-Numpy/CSV_AirplaneDelays/AirplaneDelays/DelayedFlights.csv')
```

```
In [4]: # Explorando el nombre de cada columna y el tipo de datos que se encuentran en ellas.
data.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 [5]: data
```

	Unnamed: 0	Year	Month	DayOfMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	...	TaxiIn	TaxiOut	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NAS
	0	0	2008	1	3	4	2003.0	1955	2211.0	2225	WN	...	4.0	8.0	0	N	0	NaN	NaN
	1	1	2008	1	3	4	754.0	735	1002.0	1000	WN	...	5.0	10.0	0	N	0	NaN	NaN
	2	2	2008	1	3	4	628.0	620	804.0	750	WN	...	3.0	17.0	0	N	0	NaN	NaN
	3	4	2008	1	3	4	1829.0	1755	1959.0	1925	WN	...	3.0	10.0	0	N	0	2.0	0.0
	4	5	2008	1	3	4	1940.0	1915	2121.0	2110	WN	...	4.0	10.0	0	N	0	NaN	NaN
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	1936753	7009710	2008	12	13	6	1250.0	1220	1617.0	1552	DL	...	9.0	18.0	0	N	0	3.0	0.0
	1936754	7009717	2008	12	13	6	657.0	600	904.0	749	DL	...	15.0	34.0	0	N	0	0.0	57.0
	1936755	7009718	2008	12	13	6	1007.0	847	1149.0	1010	DL	...	8.0	32.0	0	N	0	1.0	0.0
	1936756	7009726	2008	12	13	6	1251.0	1240	1446.0	1437	DL	...	13.0	13.0	0	N	0	NaN	NaN
	1936757	7009727	2008	12	13	6	1110.0	1103	1413.0	1418	DL	...	8.0	11.0	0	N	0	NaN	NaN

1936758 rows × 30 columns

```
In [6]: # visualitzar Les columnes ocultades al mig, Les de [10 a 20], que no podem veure a la taula general de dalt.
data.iloc[:, 10:20]
```

Out[6]:

	FlightNum	TailNum	ActualElapsedTime	CRSElapsedTime	AirTime	ArrDelay	DepDelay	Origin	Dest	Distance	
	0	335	N712SW	128.0	150.0	116.0	-14.0	8.0	IAD	TPA	810
	1	3231	N772SW	128.0	145.0	113.0	2.0	19.0	IAD	TPA	810
	2	448	N428WN	96.0	90.0	76.0	14.0	8.0	IND	BWI	515
	3	3920	N464WN	90.0	90.0	77.0	34.0	34.0	IND	BWI	515
	4	378	N726SW	101.0	115.0	87.0	11.0	25.0	IND	JAX	688
	...	...	...	...	...	...	...	...	...	...	...
	1936753	1621	N938DL	147.0	152.0	120.0	25.0	30.0	MSP	ATL	906
	1936754	1631	N3743H	127.0	109.0	78.0	75.0	57.0	RIC	ATL	481
	1936755	1631	N909DA	162.0	143.0	122.0	99.0	80.0	ATL	IAH	689
	1936756	1639	N646DL	115.0	117.0	89.0	9.0	11.0	IAD	ATL	533
	1936757	1641	N908DL	123.0	135.0	104.0	-5.0	7.0	SAT	ATL	874

1936758 rows × 10 columns

1.2 Explica breument quines variables hi ha.

Descripció general de les columnes del dataset i el tipus de dada (Dtype) per cada columna :

- hi ha 30 columnes de, [0] a [29]
- hi ha 1.936.758 rangs (entrades)
- les dades sont totes recopilades de vols de l'any 2008 als Estats-Units
- hi han valors només dels següents : float64 (14), int64 (11), object (5)
- a les columnes Year [1], Month [2], DayofMonth [3], DayOf Week [4], es troben la data per any, mes i dia dels vols
- a les columnes DepTime [5], CRSDepTime [6], ArrTime [7], CRSArrTime [8], es troben les hores de començament i fi dels vols, amb l'índice "CRS" indicant l'hora planificada i l'altra variable es la real.
- la columna UniqueCarrier [9] té com a tipus de dades "objectes" en forma del letres, que son el símbol de la companya aèria (aerolinia)
- la columna TailNum [11] conté el número de matrícula o "registration number" únic a cada aeronau
- les columnes [15] a [16] indiquen els temps de duració dels viatge, sient AirTime [14] el temps en minuts passats en el aire
- CRSElapsedTime [13] es el temps total del viatge (terra+aire) previst en minuts i ActualElapsedTime [12] el el temps real total del vol.
- la columna Origin [17] i Dest [18] indiquen aeroports d'inici i destinació
- la columna Distance [19] és la distancia en milles del recorregut
- les columnes de TaxiIn [21] i TaxiOut [22] són gestions que potser que no ens interessin per al nostre anàlisi
- les columnes [23 i 24] indiquen si els viatges ha sigut cancel·lats o desviats
- les columnes [25 a 29] mostren dades en minuts i per cada motiu de retard ; aquí n'hi han moltes dades manquants o imputades de 2 maneres diferents (exemples : NaN, 0.0, 32.0, etc).
- Info sobre NASDelays [27]: " delays or cancellations coded "NAS" are the type of weather delays that could be reduced with corrective action by the airports or the Federal Aviation Administration".

Aquí podem veure quantes dates úniques n'hi han per columna que ens ajuden a decidir quines columnes mantindre i quines esborrar. Destaquem :

- hi ha 1 sol valor a la columna any, que es el 2008
- hi han 20 valors corresponents als codis d'aerolínies
- hi ha 303 eroports d'origen i 304 d'arribada
- les columnes \*Cancelled\* i \*CancellationCode\* no són equivalents donat que la primera mostra 2 valors únics i la segona, 4, aixi doncs no esborrarem cap d'elles de moment.

In [7]:

```
# veure quantes dates úniques n'hi han per columna.
data.nunique()
```

Out[7]:

```
Unnamed: 0      1936758
Year            1
Month           12
DayofMonth      31
DayOfWeek        7
DepTime         1438
CRSDepTime      1207
ArrTime         1440
CRSArrTime      1364
UniqueCarrier   20
FlightNum       7499
TailNum         5366
ActualElapsedTime 673
CRSElapsedTime  515
AirTime         650
ArrDelay        1128
DepDelay        1058
Origin          303
Dest            304
Distance        1419
TaxiIn          180
TaxiOut         332
Cancelled        2
CancellationCode  4
Diverted         2
CarrierDelay     983
WeatherDelay     599
NASDelay         574
SecurityDelay    156
LateAircraftDelay 564
dtype: int64
```

1.3 Queda't únicament amb les columnes que consideris rellevants, justificant la teva elecció.

- Quitar del df original las columnas que no son necesarias: *Unnamed: 0*, *Year*, *AirTime*, *TaxiIn*, *TaxiOut*, porque no influyen en los cálculos posteriores sobre tiempos de retraso, compaías o momentos clave. El año es siempre 2008.
  - En lo que respeta las de los diferentes tipos de *Delays* (causas de retraso), *CancellationCode* y *Diverted*, de momento las dejamos, pero con la intención de, posteriormente en el ejercicio 2, unir sus datos en una sola columna.
  - Se mantienen las variables que puedan ayudar a detectar momentos y lugares clave, patrones en las aerolíneas, trayectos, distancias, incluso el tipo / número de matrícula individual de cada aeronave.
  - les columnes *Cancelled* i *CancellationCode* no són equivalents donat que la primera mostra 2 valors únics i la segona, 4, aixi doncs les mantenim de moment.

In [8]:

```
# Esborrem algunes columnes :
new_df=data.drop(columns=["Unnamed: 0", "Year", "AirTime","TaxiIn", "TaxiOut"], axis=1)
```

In [9]:

```
# veiem de nou els tipus de valors i els noms de cada columna
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1936758 entries, 0 to 1936757
Data columns (total 25 columns):
#   Column                Dtype
---  -
0   Month                 int64
1   DayOfMonth            int64
2   DayOfWeek             int64
3   DepTime               float64
4   CRSDepTime            int64
5   ArrTime               float64
6   CRSArrTime            int64
7   UniqueCarrier         object
8   FlightNum             int64
9   TailNum               object
10  ActualElapsedTime     float64
11  CRSElapsedTime        float64
12  ArrDelay               float64
13  DepDelay               float64
14  Origin                 object
15  Dest                   object
16  Distance               int64
17  Cancelled              int64
18  CancellationCode      object
19  Diverted               int64
20  CarrierDelay           float64
21  WeatherDelay           float64
22  NASDelay               float64
23  SecurityDelay          float64
24  LateAircraftDelay      float64
dtypes: float64(11), int64(9), object(5)
memory usage: 369.4+ MB
```

- Quan la columna és completa, es mostren 1936758 entrades
- El recompte dels valors no nuls ens mostra on hi ha valors manquants :
- "ArrTime" [5], TailNum [9], ActualElapsedTime [10], CRSElapsedTime [11], ArrDelay [12], CarrierDelay [20], WeatherDelay [21], NASDelay [22], SecurityDelay [23], LateAircraftDelay [24]

```
In [10]: # Recompte dels valors "non-NA" o no nuls a les columnes del nou dataframe
```

```
new_df.count()
```

```
Out[10]:
Month                1936758
DayOfMonth           1936758
DayOfWeek             1936758
DepTime              1936758
CRSDepTime           1936758
ArrTime              1929648
CRSArrTime           1936758
UniqueCarrier        1936758
FlightNum            1936758
TailNum              1936753
ActualElapsedTime    1928371
CRSElapsedTime       1936560
ArrDelay              1928371
DepDelay              1936758
Origin               1936758
Dest                 1936758
Distance              1936758
Cancelled             1936758
CancellationCode      1936758
Diverted              1936758
CarrierDelay          1247488
WeatherDelay          1247488
NASDelay              1247488
SecurityDelay         1247488
LateAircraftDelay     1247488
dtype: int64
```

1.4 Redueix la dimensió del dataset de manera aleatòria per tal d'obtenir un dataset de només 200.000 registres. Tots els exercicis s'han de fer amb aquest dataset reduït.

```
In [11]: # reducció del dataset
data_reduced = new_df.sample(n=200000)
```

EXERCICI 2 : Fes un informe complet del dataset:

- 2.1 Resumeix estadísticament el dataset i les columnes d'interès. Fes una anàlisi estadístic del que consideris rellevant.
- 2.2 Troba quantes dades faltants hi ha per columna.
- 2.3 Crea columnes noves (velocitat mitjana del vol, si ha arribat tard o no...).
- 2.4 Fes una taula de les aerolínies amb més endarreriments acumulats.
- 2.5 Quins són els vols més llargs? I els més endarrerits? Busca les rutes més llargues i les que acumulen més retards.
- 2.6 Aporta allò que consideris rellevant.

2.1 Resumeix estadísticament el dataset i les columnes d'interès. Fes una anàlisi estadístic del que consideris rellevant.

- Queden 25 columnes amb 200.000 rangs i la següent inforació :

```
In [12]: data_reduced.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200000 entries, 1589022 to 124973
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Month                 200000 non-null  int64
1   DayOfMonth            200000 non-null  int64
2   DayOfWeek             200000 non-null  int64
3   DepTime               200000 non-null  float64
4   CRSDepTime            200000 non-null  int64
5   ArrTime               199261 non-null  float64
6   CRSArrTime            200000 non-null  int64
7   UniqueCarrier         200000 non-null  object
8   FlightNum             200000 non-null  int64
9   TailNum               199999 non-null  object
10  ActualElapsedTime     199148 non-null  float64
11  CRSElapsedTime        199987 non-null  float64
12  ArrDelay               199148 non-null  float64
13  DepDelay              200000 non-null  float64
14  Origin                200000 non-null  object
15  Dest                  200000 non-null  object
16  Distance              200000 non-null  int64
17  Cancelled             200000 non-null  int64
18  CancellationCode      200000 non-null  object
19  Diverted              200000 non-null  int64
20  CarrierDelay          128652 non-null  float64
21  WeatherDelay          128652 non-null  float64
22  NASDelay              128652 non-null  float64
23  SecurityDelay         128652 non-null  float64
24  LateAircraftDelay     128652 non-null  float64
dtypes: float64(11), int64(9), object(5)
memory usage: 39.7+ MB
```

- Per facilitar la visualització i de proporcions a la pantalla, separem les vstes en 2 trams de dimensions similars : els de temps d'un costat i de l'altre, els retards, cancel·lacions i desviacions :

```
In [13]: data_reduced.iloc[:, 0:12]
```

	Month	DayOfMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNum	TailNum	ActualElapsedTime	CRSElapsedTime
1589022	10	17	5	1916.0	1833	2047.0	2007	FL	619	N995AT	91.0	94.0
1928457	12	15	1	2212.0	1920	40.0	2204	CO	346	N12313	148.0	164.0
913776	6	26	4	1743.0	1720	1809.0	1735	WN	2173	N521SW	86.0	75.0
861618	5	1	4	1623.0	1610	1745.0	1735	AA	2242	N291AA	82.0	85.0
1837180	12	18	4	2151.0	2134	37.0	7	DL	1063	N644DL	346.0	333.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1110168	7	30	3	1850.0	1800	2008.0	1925	WN	1656	N224WN	78.0	85.0
1016845	6	5	4	1652.0	1610	1832.0	1750	MQ	4259	N649PP	100.0	100.0
1451434	9	13	6	1205.0	947	1253.0	1049	YV	7116	N77331	108.0	122.0
306887	2	4	1	1412.0	1345	1546.0	1535	MQ	3677	N684JW	94.0	110.0
124973	1	19	6	1112.0	1045	1218.0	1145	MQ	4202	N617AE	66.0	60.0

200000 rows × 12 columns

```
In [14]: data_reduced.iloc[:, 12:25]
```

	ArrDelay	DepDelay	Origin	Dest	Distance	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay
1589022	40.0	43.0	TPA	ATL	406	0	N	0	0.0	0.0	40.0	0.0	0.0
1928457	156.0	172.0	IAH	ORD	925	0	N	0	156.0	0.0	0.0	0.0	0.0
913776	34.0	23.0	ELP	PHX	347	0	N	0	1.0	0.0	11.0	0.0	22.0
861618	10.0	13.0	SFO	LAX	337	0	N	0	NaN	NaN	NaN	NaN	NaN
1837180	30.0	17.0	ATL	SEA	2182	0	N	0	12.0	0.0	13.0	0.0	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1110168	43.0	50.0	OAK	SNA	371	0	N	0	43.0	0.0	0.0	0.0	0.0
1016845	42.0	42.0	ORD	MEM	491	0	N	0	15.0	0.0	0.0	0.0	27.0
1451434	124.0	138.0	GSO	ORD	590	0	N	0	0.0	0.0	0.0	0.0	124.0
306887	11.0	27.0	DFW	DSM	624	0	N	0	NaN	NaN	NaN	NaN	NaN
124973	33.0	27.0	LSE	ORD	215	0	N	0	10.0	0.0	6.0	0.0	17.0

200000 rows × 13 columns

Estadístiques bàsiques amb la funció describe() :

- El anàlisis amb la funció bàsica "describe", podem veure que aquestes estadístiques son rellevants en les dades en dies de l'any, o en minuts de demores i temps de duració, però per veure infromació de les columnes d'objectes, no ens aporten.
- Ens adonem doncs que de que en molts casos es irrellevant fer un "count", sumar i fer mitjanes simples.
- El valor mínim (min) en algunes columnes crida l'atenció ; poden ser errors, vol en proves, vols cancel·lats, avions privats, altres casos d'excepció. *FlightNum :1, ActualElapsedTime :16, CRSElapsedTime : 1, ArrDelay :-68.*
- Es interessant contrastar la mitjana simple "mean" amb el 50% percentil, ja que en principi donen dades més o menys properes a les columnes *Month, DayOfMonth, DayOfWeek, DepTime, CRSDepTime, ArrTime, CRSArrTime* i entre *Cancelled* i *Diverted*, però a partir de *ActualElapsedTime, CRSElapsedTime, ArrDelay, DepDelay, Distance* divergen molt i més encara quan observem les temps (min) per cada causa de retards amb *CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay*
- Sobre això :
  - > la mitjana simple (mean) és sensible als valors extrems, en canvi, els 50 dona una idea d'on es troben el 50% o la meitat dels valors i presenta números molt més baixos que la mitjana simple.
  - > en el cas dels retards per cada causa, podem observar com la mitjana simple (mean) s'aproxima més al 75 percentil que al 50, volant dir que pocs valors excepcionalment grans influencien les estadístiques del dataset.

```
In [15]: data_reduced.describe()
```

Out[15]:

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	FlightNum	ActualElapsedTime	CRSElapsedTime	ArrDelay	DepDelay	Distance	C
count	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	199261.000000	200000.000000	200000.000000	199148.000000	199987.000000	199148.000000	200000.000000	200000.000000	20000
mean	6.101245	15.751845	3.981760	1518.205585	1467.525630	1610.650719	1634.488025	2178.913630	133.457499	134.517109	41.862625	42.888155	767.185370	
std	3.478620	8.778046	1.997781	450.299468	424.678981	548.284399	465.317273	1941.212918	72.124724	71.440591	56.379571	52.939936	574.690811	
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	16.000000	-18.000000	-66.000000	6.000000	30.000000	
25%	3.000000	8.000000	2.000000	1202.000000	1135.000000	1316.000000	1325.000000	610.000000	80.000000	82.000000	9.000000	12.000000	340.000000	
50%	6.000000	16.000000	4.000000	1545.000000	1510.000000	1715.000000	1705.000000	1540.000000	116.000000	117.000000	24.000000	24.000000	610.000000	
75%	9.000000	23.000000	6.000000	1900.000000	1817.000000	2031.000000	2015.000000	3411.000000	165.000000	165.000000	55.000000	53.000000	998.000000	
max	12.000000	31.000000	7.000000	2400.000000	2359.000000	2400.000000	2400.000000	9740.000000	1114.000000	660.000000	1951.000000	1952.000000	4962.000000	

A més d'aquestes estadístiques bàsiques, provem d'analitzar el dataset amb la funció estadística del valor de MODA ("mode"), on veiem quin es cas més freqüent per cada variable

- Per el dataset que ens interessa es una informació rellevant donat que ens mostra dades recurrents, moments de l'any i del dia on es concentren els vols, els retards, vols, amb quina companya, quin es el dia i l'hora més habituals (més circulació d'avions als aeroports) ; també ens dona una idea de si lo habitual es que els avions surtin a temps, o al contrari, que tinguin retards (concretament a les últimes columnes)
- data.mode() ens dona tots els resultats a la fila de dalt (0) i deixa les altres buides. Per tant, li demanem "data.mode().head(1)"

In [16]:

```
data_reduced.iloc[:, 0:12].mode().head(1)
```

Out[16]:

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNum	TailNum	ActualElapsedTime	CRSElapsedTime
0	6.0	22.0	5.0	1800.0	1800.0	2045.0	1930.0	WN	50.0	N612SW	75.0	75.0

In [17]:

```
data_reduced.iloc[:, 12:25].mode().head(1)
```

Out[17]:

	ArrDelay	DepDelay	Origin	Dest	Distance	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay
0	9.0	6.0	ATL	ORD	337	0	N	0	0.0	0.0	0.0	0.0	0.0

- Primeres observacions bàsiques però útils amb MODE :
  - l'hora d'arribada real a destinació (ArrTime) que més surt és 20:30 ArrTime (amb el dataset original el mateix càlcul donava 21:00)
  - la duració més habitual d'aquesta llista de vols és de 75 minuts (amb el dataset original el mateix càlcul donava 80 min)
  - lo més habitual es que els vols no es calcel·lin, ni que siguin desviats
  - la companya aèria més utilitzada per a aquest dataset és WN (Southwest Airlines)
  - l'aeroport de sortida que més vegades apareix és Atlanta (ATL) i el de destí, Orlando (ORD)
  - el vol número 16 (molt probablement WN16) es el que més vols efectua i apareix més vegades al dataset
  - les el temps de retard més habitual sol ser de 10 minuts a l'arribada i de 6 minuts a la sortida dels vols
  - la distància més freqüent es de 337 (km o miles)
  - lo més habitual es que els avions surtin bastant a temps i com previst, tenint en compte que si el retard es menor a 15 minuts, no compta com a retard

- MODE també ens dona pistes sobre el tràfic a les pistes i les portes dels diferents aeroports:

- el mes de l'any on hi han més vols es desembre (mes 12)
- el dia del mes amb més sortides ("departures") sol ser el 22,i el de la setmana, el divendres (dia 5)
- l'hora de sortida de més afluència de vols (no necessàriament de persones ja que depén del tamany de cada aeronau) tant la prevista = real de és a les 18:00
- l'hora d'arribada planificada més freqüentment per les aerolínies es a les 19:30 del vespre, mentres que la realitat s'aproxima més cap a les 21:00 de la nit.

Aixó podria indicar que els viatgers que planifiquen volar pel vespre tenen bastants probabilitats d'arribar tard al seu destí.

## 2.2 Troba quantes dades faltants hi ha per columna.

In [18]:

```
# Aquí veiem quants valors nuls hi ha a Les columnes del nou dataframe, des de Les columnes més complertes a Les que ho són menys.
```

```
null_counts = data_reduced.isnull().sum()
null_counts[null_counts > 0].sort_values(ascending=True)
```

Out[18]:

TailNum	1
CRSElapsedTime	13
ArrTime	739
ActualElapsedTime	852
ArrDelay	852
CarrierDelay	71348
WeatherDelay	71348
NASDelay	71348
SecurityDelay	71348
LateAircraftDelay	71348
dtype:	int64

## 2.3 Crea columnes noves (velocitat mitjana del vol, si ha arribat tard o no...).

Les noves columnes *AverageSpeed* i *LateCode* es poden veure a la part dreta del dataset "data\_reduced"

La velocitat mitjana del vol es calcula en x miles/hour => tindrém que fer una conversió -> Distance / ActualElapsedTime són miles/minut

- > passar de miles per minut a miles per hora -> resultat \*60 min

In [48]:

```
# Creació de La columna de velocitat mitjana AverageSpeed per cada viatge.
data_reduced['AverageSpeed'] = (data_reduced['Distance']/data_reduced['ActualElapsedTime'])*60
data_reduced['AverageSpeed'] = data_reduced['AverageSpeed'].round(1)
```

In [47]:

```
# Creació de La columna del nivells d'enderririment "LateCode" per cada viatge.
# crearem un codi que digui si ha arribat a temps (0), o en cas d'arribar tard, de quina magnitud es La demora (del 1 al 3)
```

```
LateCode=[]
for x in data_reduced['ArrDelay']:
    if x > 60 :
        LateCode.append(3)
    elif x > 30 :
        LateCode.append(2)
    elif x > 15 :
        LateCode.append(1)
    else:
        LateCode.append(0)
```

```
data_reduced['LateCode']=LateCode
data_reduced
```

Out[47]:

	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNum	TailNum	...	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	Securi
1589022	10	17	5	1916.0	1833	2047.0	2007	FL	619	N995AT	...	0		N	0	0.0	0.0	40.0
1928457	12	15	1	2212.0	1920	40.0	2204	CO	346	N12313	...	0		N	0	156.0	0.0	0.0
913776	6	26	4	1743.0	1720	1809.0	1735	WN	2173	N521SW	...	0		N	0	1.0	0.0	11.0
861618	5	1	4	1623.0	1610	1745.0	1735	AA	2242	N291AA	...	0		N	0	NaN	NaN	NaN
1837180	12	18	4	2151.0	2134	37.0	7	DL	1063	N644DL	...	0		N	0	12.0	0.0	13.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1110168	7	30	3	1850.0	1800	2008.0	1925	WN	1656	N224WN	...	0		N	0	43.0	0.0	0.0
1016845	6	5	4	1652.0	1610	1832.0	1750	MQ	4259	N649PP	...	0		N	0	15.0	0.0	0.0
1451434	9	13	6	1205.0	947	1253.0	1049	YV	7116	N77331	...	0		N	0	0.0	0.0	0.0
306887	2	4	1	1412.0	1345	1546.0	1535	MQ	3677	N684JW	...	0		N	0	NaN	NaN	NaN
124973	1	19	6	1112.0	1045	1218.0	1145	MQ	4202	N617AE	...	0		N	0	10.0	0.0	6.0

200000 rows × 27 columns



In [81]:

```
# Les dades de les columnes per comprovació:  
data_reduced['AverageSpeed'].head(5), data_reduced['LateCode'].head(5)
```

Out[81]:

```
(1589022    267.7  
1928457    375.0  
913776     242.1  
861618     246.6  
1837180    378.4  
Name: AverageSpeed, dtype: float64,  
1589022      2  
1928457      3  
913776        2  
861618        0  
1837180        1  
Name: LateCode, dtype: int64)
```

2.4 Fes una taula de les aerolínies amb més endarreriments acumulats.

Primer obtenim només els vols del dataset que no tinguin endarreriments "oficials" (no més de 15 min)

-> ho fem filtrant només els vols amb codi LateCode que no sigui 0, (on 0 es que ha arribat a l'hora)

In [241]:

```
VuelosRetrasados = data_reduced[(data_reduced.LateCode > 0)]
```

En resum, les aerolínies menys puntuals en numero de vols (indepentdentment del seu tamany), de més a menys

-> l'aerolinia amb més retards és Sowthwest Airlines (WN) amb 20308 registres, la que també va apareixer en les estadístiques inicials amb "mode".

-> la segona és American Airlines (AA) i la tercera, Envoy Air (MQ).

In [62]:

```
CarrierDelayed=VuelosRetrasados[['UniqueCarrier','LateCode']].groupby(['UniqueCarrier']).count()  
CarrierDelayed.sort_values(by=['LateCode'], ascending=False)
```

Out[62]:

	LateCode
UniqueCarrier	
WN	20308
AA	13288
MQ	9850
UA	9687
OO	8715
XE	7359
DL	7231
US	5991
CO	5919
EV	5578
NW	5463
YV	5139
FL	4652
OH	4014
B6	3995
9E	3609
AS	2455
F9	1596
HA	438
AQ	21

2.5 Quins són els vols més endarrerits? Els més llargs? Busca les rutes més llargues i les que acumulen més retards.

Els vols que MÉS SOVINT s'han endarrerit són els vols número 50, 55, 40, 65, i 44.

In [112]:

```
Vols=VuelosRetrasados[['FlightNum','LateCode']].groupby(['FlightNum']).count()  
Vols.sort_values(by=['LateCode'], ascending=False)
```

Out[112]:

LateCode	
FlightNum	
50	121
55	99
40	99
65	98
44	97
...	...
7232	1
4905	1
4904	1
6401	1
9740	1

7205 rows × 1 columns

En canvi, nos son els mateixos els vols que MÉS MINUTS DE RETARD CUMULEN en tot l'any 2008 són els vols número 50, 15, 40, 29, i 75.

```
In [120]: VolsMinDelay=VuelosRetrasados[['FlightNum', 'ArrDelay']].groupby(['FlightNum']).sum()
VolsMinDelay.sort_values(by=['ArrDelay'], ascending=False)
```

Out[120]:

ArrDelay	
FlightNum	
50	6650.0
15	6288.0
40	6234.0
29	6224.0
75	6209.0
...	...
6103	16.0
4954	16.0
6353	16.0
2392	16.0
6126	16.0

7205 rows × 1 columns

Els vols més llargs (en milles -> distancia) son els següents:

```
In [175]: Dist=data_reduced[['FlightNum', 'Distance']].groupby(['Distance']).count()
Dist.sort_values(by=['Distance'], ascending=False)
```

Out[175]:

FlightNum	
Distance	
4962	37
4502	45
4243	22
4213	3
4184	6
...	...
41	14
37	1
36	2
31	21
30	1

1368 rows × 1 columns

Els vols que han sigut més llargs en minuts de duració real, son els següents:

```
In [240]: LlargTemps=data_reduced[['FlightNum', 'ActualElapsedTime']].groupby(['FlightNum']).head(10)
LlargTemps.sort_values(by=['ActualElapsedTime'], ascending=False).head(10)
```

Out[240]:

FlightNum	ActualElapsedTime	
365249	15	663.0
1836284	1273	611.0
652101	1273	599.0
528337	73	580.0
567739	851	576.0
1075077	1881	576.0
1174491	1561	556.0
447763	3	515.0
1427947	1282	512.0
1135289	5203	505.0

## 2.6 Aporta allò que consideris rellevant.

Aqui veiem quants viatges han sigut puntuals (codi 0), i quands han tingut endarreriments, per codi de retard (1, 2, 3)

-> la taula mostra que sobre 200.000 viatges en total, hi ha hagut més viatges enderrerrits (uns 125.000) que no pas a l'hora (74.692)

-> aprox. 75.000, encara que 'a l'hora' sigui el valor amb més freqüència dels tres (com ens anava anuniant la funció "mode" al principi)

-> observem també que sobre 200.000 viatges en total, uns 85.000 van sortir al menys 30 minuts més tard de l'hora prevista.

```
In [146]: RetrasoVuelosCode=data_reduced[['UniqueCarrier', 'LateCode']].groupby(['LateCode']).count()
RetrasoVuelosCode
```

Out[146]:

UniqueCarrier	
LateCode	
0	74692
1	39997
2	40205
3	45106

Les aerolínies amb més cancel·lacions: Envoy Air (MQ) és la que més en té

```
In [60]: CarrierCancelations=Cancelations[['UniqueCarrier', 'Cancelled']].groupby(['UniqueCarrier']).count()
CarrierCancelations.sort_values(by=['Cancelled'], ascending=False)
```

Out[60]:

Cancelled	
UniqueCarrier	
MQ	12
OO	10
YV	7
EV	6
XE	5
9E	4
CO	4
UA	4
AA	3
OH	3
B6	2
DL	2
NW	2
US	2
WN	2

Els mesos de l'any amb més cancel·lacions: desembre, novembre i octubre

```
In [143]: Cancel=Cancelations[['Month', 'Cancelled']].groupby(['Month']).count()
Cancel.sort_values(by=['Cancelled'], ascending=False).head(10)
```

Out[143]:

Cancelled	
Month	
12	50
11	12
10	6

En canvi, NO son els mateixos que els mesos amb més endarreriments, que són desembre, juny, març, febrer i gener.

-> aquests són els que corresponen amb les festes de Nadal i els viatges d'estiu, del "March Break dels estudiants", de després de l'Any Nou.

```
In [245]: RetardsMesos=VuelosRetrasados[['Month', 'DepDelay']].groupby(['Month']).count()
RetardsMesos.sort_values(by=['DepDelay'], ascending=False)
```

Out[245]:

DepDelay	
Month	
12	14161
6	13863
3	13292
2	12760
1	12220
7	11947
8	10043
4	9885
5	9562
11	6418
10	5582
9	5575

Les aeronaus que més distància van recorre l'any 2008 : interessant per saber quan fer el seu manteniment

```
In [190]: Aeronaus_milles=data_reduced[['TailNum', 'ActualElapsedTime']].groupby(['TailNum']).sum()
Aeronaus_milles.sort_values(by=['ActualElapsedTime'], ascending=False)
```



Out[190]:

ActualElapsedTime	
TailNum	
N832MH	14802.0
N57855	12392.0
N612SW	12148.0
N621JB	11728.0
N57863	11696.0
...	...
N384AE	58.0
N322SW	56.0
N841AL	36.0
N843AL	26.0
9169E	0.0

5277 rows × 1 columns

Quants vols hi han sortit de cada aeroport i on hi ha hagut més :

In [210]:

```
Vols_per_aeroport= (
    data_reduced
    .groupby(['Origin'])
    .apply(lambda x: pd.Series({
        'nb_flights': len(x['FlightNum'])
    })))
Vols_per_aeroport.sort_values(by=['nb_flights'], ascending=False)
```

Out[210]:

nb_flights	
Origin	
ATL	13366
ORD	13075
DFW	9960
DEN	7762
LAX	6011
...	...
YKM	2
SUX	1
INL	1
CMX	1
PUB	1

298 rows × 1 columns

Donat que la major part dels endarreriments es donen en origen (veure dataframe original), veiem els aeroports de sorida on els avions triguen més en sortir:

-> mostra els codis dels aeroports que presenten més retards (comptabilitzats des dels minuts) a les sortides dels avions

-> **Orlando, Atlanta, Washington DF, Denver.**

-> on més endarreriments hi ha és als aeroports amb més avions en circulació

In [68]:

```
AeroportsRetras=VuelosRetrasados[['Origin','DepDelay']].groupby(['Origin']).count()
AeroportsRetras.sort_values(by=['DepDelay'], ascending=False)
```

Out[68]:

DepDelay	
Origin	
ORD	9269
ATL	8638
DFW	6276
DEN	4687
EWR	3749
...	...
SUX	1
CMX	1
BPT	1
YKM	1
PUB	1

296 rows × 1 columns

### Exercici 3. Exporta el dataset a Excel

In [247]:

```
data_reduced.to_excel("data_reduced_delays.xlsx")
```