S03T05

November 11, 2021

1 Nivell 1

1.1 - 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.

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv (r'DelayedFlights.csv',index_col=0)
df.columns
```

[ ]:
```
Index(['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')
```

Definicio dels camps 1. **Year** -> any de la dada format yyyy 2. **Month** -> mes de la dada format mm 3. **DayofMonth** -> dia del mes 1-31 4. **DayOfWeek** -> dia de la setmana 1 (Monday) - 7 (Sunday) 5. **DepTime** -> hora de sortida (local, hhmm) 6. **CRSDepTime** -> hora programada de sortida (local, hhmm) 7. **ArrTime** -> hora de arrivada (local, hhmm) 8. **CRSArrTime** -> hora programda de arrivada (local, hhmm) 9. **UniqueCarrier** -> identificador del operador 10. **FlightNum** -> numero de vol 11. **TailNum** -> matricula del avio 12. **ActualElapsedTime** -> temps de vol total en minuts 13. **CRSElapsedTime** -> temps estimat de vol total en minutos 14. **AirTime** -> temps en el aire en minuts 15. **ArrDelay** -> Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers, in minutes 16. **DepDelay** -> Difference in minutes between scheduled and actual departure time.Early departures show negative numbers, in minutes 17. **Origin** -> codi IATA areoport de origen 18. **Dest** -> codi IATA aeroport de dest 19. **Distance** -> distancia entre aeroports (miles) 20. **TaxiIn** -> Wheels down and arrival at the destination airport gate, in minutes 21. **TaxiOut** -> The time elapsed between departure from the origin airport gate and wheels off, in minutes 22. **Cancelled** -> vol cancelat o no 23. **CancellationCode** -> motiu de la cancelacio (A = carrier, B = weather, C = NAS, D = security) 24. **Diverted** -> Desviat 1 = yes, 0 = no 25. **CarrierDelay** -> Retràs degut a l'operador in minutes 26. **WeatherDelay** -> Retràs degut al temps in minutes 27. **NASDelay** -> Retràs degut a NAS in minutes 28. **SecurityDelay** -> Retràs degut motius de seguretat in minutes 29. **LateAircraftDelay** -> Retràs acumulat de l'avió in minutes

[ ]:
```
df.head()
```

[ ]:
```
   Year  Month  DayofMonth  DayOfWeek  DepTime  CRSDepTime  ArrTime  \
0  2008      1           3          4   2003.0        1955   2211.0
1  2008      1           3          4    754.0         735   1002.0
2  2008      1           3          4    628.0         620    804.0
4  2008      1           3          4   1829.0        1755   1959.0
5  2008      1           3          4   1940.0        1915   2121.0

   CRSArrTime UniqueCarrier  FlightNum  … TaxiIn  TaxiOut  Cancelled  \
0        2225            WN        335  …    4.0      8.0          0
1        1000            WN       3231  …    5.0     10.0          0
2         750            WN        448  …    3.0     17.0          0
4        1925            WN       3920  …    3.0     10.0          0
5        2110            WN        378  …    4.0     10.0          0

   CancellationCode  Diverted  CarrierDelay WeatherDelay NASDelay  \
0                 N         0           NaN          NaN      NaN
1                 N         0           NaN          NaN      NaN
2                 N         0           NaN          NaN      NaN
4                 N         0           2.0          0.0      0.0
5                 N         0           NaN          NaN      NaN

   SecurityDelay  LateAircraftDelay
0            NaN                NaN
1            NaN                NaN
2            NaN                NaN
4            0.0               32.0
5            NaN                NaN

[5 rows x 29 columns]
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1936758 entries, 0 to 7009727
Data columns (total 16 columns):
 #   Column          Dtype
---  ------          -----
 0   Year            int64
 1   Month           int64
 2   DayofMonth      int64
 3   DepTime         float64
 4   ArrTime         float64
```

```
 5   UniqueCarrier  object
 6   FlightNum      int64
 7   AirTime        float64
 8   ArrDelay       float64
 9   Origin         object
 10  Dest           object
 11  Distance       int64
 12  DepDate        datetime64[ns]
 13  AvgSpeed       float64
 14  Delayed        bool
 15  totalTime      float64
dtypes: bool(1), datetime64[ns](1), float64(6), int64(5), object(3)
memory usage: 302.8+ MB
```

[ ]:
```
df.count().sort_values()
```

[ ]:
```
AirTime          1928371
ArrDelay         1928371
AvgSpeed         1928371
ArrTime          1929648
totalTime        1929648
Year             1936758
Month            1936758
DayofMonth       1936758
DepTime          1936758
UniqueCarrier    1936758
FlightNum        1936758
Origin           1936758
Dest             1936758
Distance         1936758
DepDate          1936758
Delayed          1936758
dtype: int64
```

Vemos que la cantidad de valores de las 5 primeras columnas es inferior al total, por lo que implica valores NaN

[ ]:
```
df.drop(df.columns.difference(['ArrDelay', 'FlightNum', 'UniqueCarrier',␣
 ↪'AirTime',  'Distance','Origin','Dest', 'Year', 'Month',␣
 ↪'DayofMonth','ArrTime','DepTime' ]),axis=1,  inplace=True)
```

1.2 - 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.

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

```
[ ]: df['DepDate'] = pd.to_datetime(df.Year*10000+df.Month*100+df.
     ↪DayofMonth,format='%Y%m%d')
     df['AvgSpeed'] = round(60*df.Distance/df.AirTime,2) # milles per hora
     df['Delayed'] = df.ArrDelay >0
```

```
[ ]: df.head()
```

```
[ ]:    Year  Month  DayofMonth  DepTime  ArrTime UniqueCarrier  FlightNum  \
     0  2008      1           3   2003.0   2211.0            WN        335
     1  2008      1           3    754.0   1002.0            WN       3231
     2  2008      1           3    628.0    804.0            WN        448
     4  2008      1           3   1829.0   1959.0            WN       3920
     5  2008      1           3   1940.0   2121.0            WN        378

        AirTime  ArrDelay Origin Dest  Distance    DepDate  AvgSpeed  Delayed
     0    116.0     -14.0    IAD  TPA       810 2008-01-03    418.97    False
     1    113.0       2.0    IAD  TPA       810 2008-01-03    430.09     True
     2     76.0      14.0    IND  BWI       515 2008-01-03    406.58     True
     4     77.0      34.0    IND  BWI       515 2008-01-03    401.30     True
     5     87.0      11.0    IND  JAX       688 2008-01-03    474.48     True
```

Dades faltants per columna

```
[ ]: df[df.columns[df.isnull().sum(axis = 0)>0]].isnull().sum(axis=0)
```

```
[ ]: ArrTime     7110
     AirTime     8387
     ArrDelay    8387
     AvgSpeed    8387
     dtype: int64
```

Taula de les aerolínies 10 amb més endarreriments acumulats

```
[ ]: df.groupby(["UniqueCarrier"]).ArrDelay.sum().sort_values(ascending=False).
     ↪head(10)
```

```
[ ]: UniqueCarrier
     WN    11319092.0
     AA     8889066.0
     UA     6733013.0
     MQ     6396704.0
     OO     5978936.0
     XE     5176042.0
     DL     4535644.0
     CO     4045932.0
     EV     3888131.0
     YV     3691461.0
     Name: ArrDelay, dtype: float64
```

Quins són els vols més llargs en distancia

[ ]:
```
df.sort_values(by="Distance",ascending=False).
↪head(10)[['FlightNum','UniqueCarrier','Origin','Dest','Distance']]
```

[ ]:

	FlightNum	UniqueCarrier	Origin	Dest	Distance
4200196	14	CO	HNL	EWR	4962
6979519	14	CO	HNL	EWR	4962
2353671	15	CO	EWR	HNL	4962
6982535	14	CO	HNL	EWR	4962
566426	15	CO	EWR	HNL	4962
6982536	15	CO	EWR	HNL	4962
2951746	15	CO	EWR	HNL	4962
566384	15	CO	EWR	HNL	4962
6979520	15	CO	EWR	HNL	4962
2364843	15	CO	EWR	HNL	4962

Vols mes llargs per temps

[ ]:
```
df['totalTime']=df.ArrTime-df.DepTime
df.sort_values(by="totalTime",ascending=False).
↪head(10)[['FlightNum','UniqueCarrier','Origin','Dest','Distance','totalTime']]
```

[ ]:

	FlightNum	UniqueCarrier	Origin	Dest	Distance	totalTime
6539523	3601	WN	PHX	LAS	256	2397.0
3940676	4340	EV	ATL	PFN	247	2392.0
6082368	6563	OH	ATL	BHM	134	2385.0
4215806	1759	DL	ATL	BHM	134	2377.0
3931106	1759	DL	ATL	BHM	134	2357.0
6615130	6563	OH	ATL	BHM	134	2354.0
3927916	1002	DL	ATL	BNA	214	2351.0
3939645	4309	EV	ATL	DHN	171	2348.0
893950	1634	DL	ATL	HSV	151	2348.0
2378950	775	DL	ATL	BHM	134	2347.0

Vols mes endarrerits

[ ]:
```
df.sort_values(by="ArrDelay",ascending=False).
↪head(10)[['FlightNum','UniqueCarrier','Origin','Dest','ArrDelay']]
```

[ ]:

	FlightNum	UniqueCarrier	Origin	Dest	ArrDelay
1018798	808	NW	HNL	MSP	2461.0
2235378	1699	NW	CLT	MSP	2453.0
2832617	1107	NW	RSW	DTW	1951.0
3387883	3538	MQ	LIT	DFW	1707.0
6857047	357	NW	BOS	MSP	1655.0
5232546	512	NW	OMA	MSP	1583.0
2232494	1472	NW	MOT	MSP	1542.0
527950	2398	AA	EGE	MIA	1525.0

```
4061361         804             NW     SEA  MSP    1510.0
1634129        1743             NW     BNA  MEM    1490.0
```

1.3 - Exercici 3

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

```
[ ]: df.to_excel('./myDataFrame.xlsx', sheet_name='Sheet1')
```