Tasca 5 -

August 23, 2021

1 ITAcademy - Data Science amb Python

1.1 Sprint 3, Tasca 5: Tècniques d'exploració de dades amb Pandas

2 Airlines Delay

2.1 Airline on-time statistics and delay causes (2008)

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled and diverted flights appears in DOT's monthly Air Travel Consumer Report, published about 30 days after the month's end, as well as in summary tables posted on this website. BTS began collecting details on the causes of flight delays in June 2003. Summary statistics and raw data are made available to the public at the time the Air Travel Consumer Report is released.

Aquest Dataset està compost per les següents variables:

1. **Year**: 2008
2. **Month**: 1-12
3. **DayofMonth**: 1-31
4. **DayOfWeek**: 1 (Dilluns) - 7 (Diumenge)
5. **DepTime**: Hora de sortida real (local, hhmm)
6. **CRSDepTime**: Hora de sortida programada (local, hhmm)
7. **ArrTime**: Hora d'arribada real (local, hhmm)
8. **CRSArrTime**: Hora d'arribada programada (local, hhmm)
9. **UniqueCarrier**: Codi d'operador únic
10. **FlightNum**: Número de vol
11. **TailNum**: Matrícula de l'avió
12. **ActualElapsedTime**: Temps transcorregut real (en minuts)
13. **CRSElapsedTime**: Temps transcorregut programat (en minuts)
14. **AirTime**: Temps en l'aire (en minuts)
15. **ArrDelay**: Retràs en l'arribada (en minuts; [*1]
16. **DepDelay**: Retràs en la sortida (en minuts)
17. **Origin**: Codi IATA de l'aeroport d'origen
18. **Dest**: Codi IATA de l'aeroport de destí
19. **Distance**: Distància (en milles)
20. **TaxiIn**: Rodatge a pista (en minuts)
21. **TaxiOut**: Rodatge a porta (en minuts)
22. **Cancelled**: Si el vol ha sigut o no cancel · lat

23. **CancellationCode**: Codi amb el motiu de la cancel · lació (A = operadora, B = clima, C = NAS, D = seguretat)
24. **Diverted**: Desviat (1 = si, 0 = no)
25. **CarrierDelay**: Retràs degut a l'operador (en minuts) [*2]
26. **WeatherDelay**: Retràs degut al clima (en minuts): [*3]
27. **NASDelay**: Retràs degut al NAS (en minuts) [*4]
28. **SecurityDelay** Retràs degut a Seguretat (en minuts) [*5]
29. **LateAircraftDelay** Retràs acumulat de l'avió (en minuts) [*6]

[*1] "A flight is counted as"on time" if it operated less than 15 minutes later the scheduled time shown in the carriers' Computerized Reservations Systems (CRS)")

[*2] "Carrier delay is within the control of the air carrier. Examples of occurrences that may determine carrier delay are: aircraft cleaning, aircraft damage, awaiting the arrival of connecting passengers or crew, baggage, bird strike, cargo loading, catering, computer, outage-carrier equipment, crew legality (pilot or attendant rest), damage by hazardous goods, engineering inspection, fueling, handling disabled passengers, late crew, lavatory servicing, maintenance, oversales, potable water servicing, removal of unruly passenger, slow boarding or seating, stowing carry-on baggage, weight and balance delays."

[*3] "Weather delay is caused by extreme or hazardous weather conditions that are forecasted or manifest themselves on point of departure, enroute, or on point of arrival."

[*4] "Delay that is within the control of the National Airspace System (NAS) may include: non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc."

[*5] "Security delay is caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas."

[*6] "Arrival delay at an airport due to the late arrival of the same aircraft at a previous airport. The ripple effect of an earlier delay at downstream airports is referred to as delay propagation."

[22]:
```
import numpy as np
import pandas as pd

pd.set_option('display.max_columns', None)
pd.options.display.float_format = '{:,.2f}'.format
```

[23]:
```
df = pd.read_csv("DelayedFlights.csv", index_col=0)
df.head()
```

```
C:\Users\HP\anaconda3\lib\site-packages\numpy\lib\arraysetops.py:583:
FutureWarning: elementwise comparison failed; returning scalar instead, but in
the future will perform elementwise comparison
  mask |= (ar1 == a)
```

[23]:

	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	\
0	2008	1	3	4	2,003.00	1955	2,211.00	
1	2008	1	3	4	754.00	735	1,002.00	
2	2008	1	3	4	628.00	620	804.00	

```
4  2008      1           3           4 1,829.00        1755 1,959.00
5  2008      1           3           4 1,940.00        1915 2,121.00

   CRSArrTime UniqueCarrier  FlightNum TailNum  ActualElapsedTime  \
0        2225            WN        335  N712SW             128.00
1        1000            WN       3231  N772SW             128.00
2         750            WN        448  N428WN              96.00
4        1925            WN       3920  N464WN              90.00
5        2110            WN        378  N726SW             101.00

   CRSElapsedTime  AirTime  ArrDelay  DepDelay Origin Dest  Distance  TaxiIn  \
0          150.00   116.00    -14.00      8.00    IAD  TPA       810    4.00
1          145.00   113.00      2.00     19.00    IAD  TPA       810    5.00
2           90.00    76.00     14.00      8.00    IND  BWI       515    3.00
4           90.00    77.00     34.00     34.00    IND  BWI       515    3.00
5          115.00    87.00     11.00     25.00    IND  JAX       688    4.00

   TaxiOut  Cancelled CancellationCode  Diverted  CarrierDelay  WeatherDelay  \
0     8.00          0                N         0           NaN           NaN
1    10.00          0                N         0           NaN           NaN
2    17.00          0                N         0           NaN           NaN
4    10.00          0                N         0          2.00          0.00
5    10.00          0                N         0           NaN           NaN

   NASDelay  SecurityDelay  LateAircraftDelay
0       NaN            NaN                NaN
1       NaN            NaN                NaN
2       NaN            NaN                NaN
4      0.00           0.00              32.00
5       NaN            NaN                NaN
```

2.1.1 Exercici 1: Exploració inicial

[24]:
```
df.info(verbose=True, show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1936758 entries, 0 to 7009727
Data columns (total 29 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   Year               1936758 non-null  int64
 1   Month              1936758 non-null  int64
 2   DayofMonth         1936758 non-null  int64
 3   DayOfWeek          1936758 non-null  int64
 4   DepTime            1936758 non-null  float64
 5   CRSDepTime         1936758 non-null  int64
 6   ArrTime            1929648 non-null  float64
 7   CRSArrTime         1936758 non-null  int64
```

```
 8   UniqueCarrier      1936758 non-null  object
 9   FlightNum          1936758 non-null  int64
 10  TailNum            1936753 non-null  object
 11  ActualElapsedTime  1928371 non-null  float64
 12  CRSElapsedTime     1936560 non-null  float64
 13  AirTime            1928371 non-null  float64
 14  ArrDelay           1928371 non-null  float64
 15  DepDelay           1936758 non-null  float64
 16  Origin             1936758 non-null  object
 17  Dest               1936758 non-null  object
 18  Distance           1936758 non-null  int64
 19  TaxiIn             1929648 non-null  float64
 20  TaxiOut            1936303 non-null  float64
 21  Cancelled          1936758 non-null  int64
 22  CancellationCode   1936758 non-null  object
 23  Diverted           1936758 non-null  int64
 24  CarrierDelay       1247488 non-null  float64
 25  WeatherDelay       1247488 non-null  float64
 26  NASDelay           1247488 non-null  float64
 27  SecurityDelay      1247488 non-null  float64
 28  LateAircraftDelay  1247488 non-null  float64
dtypes: float64(14), int64(10), object(5)
memory usage: 443.3+ MB
```

[25]:
```
def missing_values_table(df):
    mis_val = df.isnull().sum()
    mis_val_percent = 100 * df.isnull().sum() / len(df)
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
    mis_val_table = mis_val_table.rename(columns = {0 : 'Valors Faltants', 1 :␣
↪'% de Valors Totals'})
    mis_val_table =  mis_val_table[mis_val_table.iloc[:,1] != 0].
↪sort_values(by='% de Valors Totals', ascending=False).round(1)
    print ("El DataFrame seleccionat conté " + str(df.shape[1]) + " columnes.\n"
        "Hi han " + str(mis_val_table.shape[0]) + " columnes amb valors␣
↪faltants.")
    if mis_val_table.shape[0] != 0:
        return mis_val_table
```

[26]:
```
missing_values_table(df)
```

```
El DataFrame seleccionat conté 29 columnes.
Hi han 13 columnes amb valors faltants.
```

[26]:

	Valors Faltants	% de Valors Totals
CarrierDelay	689270	35.60
WeatherDelay	689270	35.60
NASDelay	689270	35.60
SecurityDelay	689270	35.60

```
LateAircraftDelay          689270              35.60
ActualElapsedTime            8387               0.40
AirTime                      8387               0.40
ArrDelay                     8387               0.40
ArrTime                      7110               0.40
TaxiIn                       7110               0.40
TaxiOut                       455               0.00
CRSElapsedTime                198               0.00
TailNum                         5               0.00
```

Seleccionem del DataFrame original les següents columnes d'interés: - Month - DayOfMonth - DayOfWeek - FlightNum - DepTime - CRSDepTime - ArrTime - CRSArrTime - UniqueCarrier - ActualElapsedTime - CRSElapsedTime - AirTime - ArrDelay - DepDelay - Origin - Dest - Distance

[27]:
```
df_filtered = df[["Year", "Month", "DayofMonth", "DayOfWeek", "FlightNum",␣
↪"DepTime", "CRSDepTime", "ArrTime", "CRSArrTime", "UniqueCarrier",␣
↪"ActualElapsedTime", "AirTime", "ArrDelay", "DepDelay", "Origin", "Dest",␣
↪"Distance"]]
```

[28]:
```
df_filtered = df_filtered.assign(Distance = lambda x: x.Distance * 1.60934)␣
↪#Convert from miles to km
df_filtered = df_filtered.assign(AirTime = lambda x: x.AirTime / 60) #Convert␣
↪from minutes to hours
```

[29]:
```
df_filtered
```

[29]:

	Year	Month	DayofMonth	DayOfWeek	FlightNum	DepTime	CRSDepTime	\
0	2008	1	3	4	335	2,003.00	1955	
1	2008	1	3	4	3231	754.00	735	
2	2008	1	3	4	448	628.00	620	
4	2008	1	3	4	3920	1,829.00	1755	
5	2008	1	3	4	378	1,940.00	1915	
…	…	…	…	…	…	…	…	
7009710	2008	12	13	6	1621	1,250.00	1220	
7009717	2008	12	13	6	1631	657.00	600	
7009718	2008	12	13	6	1631	1,007.00	847	
7009726	2008	12	13	6	1639	1,251.00	1240	
7009727	2008	12	13	6	1641	1,110.00	1103	

	ArrTime	CRSArrTime	UniqueCarrier	ActualElapsedTime	AirTime	\
0	2,211.00	2225	WN	128.00	1.93	
1	1,002.00	1000	WN	128.00	1.88	
2	804.00	750	WN	96.00	1.27	
4	1,959.00	1925	WN	90.00	1.28	
5	2,121.00	2110	WN	101.00	1.45	
…	…	…	…	…	…	
7009710	1,617.00	1552	DL	147.00	2.00	
7009717	904.00	749	DL	127.00	1.30	

```
7009718 1,149.00        1010            DL             162.00     2.03
7009726 1,446.00        1437            DL             115.00     1.48
7009727 1,413.00        1418            DL             123.00     1.73

         ArrDelay  DepDelay Origin Dest  Distance
0          -14.00      8.00    IAD  TPA  1,303.57
1            2.00     19.00    IAD  TPA  1,303.57
2           14.00      8.00    IND  BWI    828.81
4           34.00     34.00    IND  BWI    828.81
5           11.00     25.00    IND  JAX  1,107.23
…               …         …      …    …         …
7009710     25.00     30.00    MSP  ATL  1,458.06
7009717     75.00     57.00    RIC  ATL    774.09
7009718     99.00     80.00    ATL  IAH  1,108.84
7009726      9.00     11.00    IAD  ATL    857.78
7009727     -5.00      7.00    SAT  ATL  1,406.56

[1936758 rows x 17 columns]
```

2.1.2 Exercici 2

2.1. Resum estadístic de les columnes d'interés

[30]:
```
df_filtered.describe(include='all')
```

[30]:
```
                Year        Month   DayofMonth    DayOfWeek    FlightNum  \
count  1,936,758.00 1,936,758.00 1,936,758.00 1,936,758.00 1,936,758.00
unique          NaN          NaN          NaN          NaN          NaN
top             NaN          NaN          NaN          NaN          NaN
freq            NaN          NaN          NaN          NaN          NaN
mean       2,008.00         6.11        15.75         3.98     2,184.26
std            0.00         3.48         8.78         2.00     1,944.70
min        2,008.00         1.00         1.00         1.00         1.00
25%        2,008.00         3.00         8.00         2.00       610.00
50%        2,008.00         6.00        16.00         4.00     1,543.00
75%        2,008.00         9.00        23.00         6.00     3,422.00
max        2,008.00        12.00        31.00         7.00     9,742.00

             DepTime   CRSDepTime      ArrTime   CRSArrTime UniqueCarrier  \
count  1,936,758.00 1,936,758.00 1,929,648.00 1,936,758.00       1936758
unique          NaN          NaN          NaN          NaN            20
top             NaN          NaN          NaN          NaN            WN
freq            NaN          NaN          NaN          NaN        377602
mean       1,518.53     1,467.47     1,610.14     1,634.22           NaN
std          450.49       424.77       548.18       464.63           NaN
min            1.00         0.00         1.00         0.00           NaN
25%        1,203.00     1,135.00     1,316.00     1,325.00           NaN
50%        1,545.00     1,510.00     1,715.00     1,705.00           NaN
```

```
75%        1,900.00     1,815.00     2,030.00     2,014.00          NaN
max        2,400.00     2,359.00     2,400.00     2,400.00          NaN

        ActualElapsedTime      AirTime     ArrDelay     DepDelay   Origin  \
count        1,928,371.00 1,928,371.00 1,928,371.00 1,936,758.00  1936758
unique                NaN          NaN          NaN          NaN      303
top                   NaN          NaN          NaN          NaN      ATL
freq                  NaN          NaN          NaN          NaN   131613
mean               133.31         1.80        42.20        43.19      NaN
std                 72.06         1.14        56.78        53.40      NaN
min                 14.00         0.00      -109.00         6.00      NaN
25%                 80.00         0.97         9.00        12.00      NaN
50%                116.00         1.50        24.00        24.00      NaN
75%                165.00         2.28        56.00        53.00      NaN
max              1,114.00        18.18     2,461.00     2,467.00      NaN

           Dest     Distance
count   1936758 1,936,758.00
unique      304          NaN
top         ORD          NaN
freq     108984          NaN
mean        NaN     1,232.25
std         NaN       924.53
min         NaN        17.70
25%         NaN       543.96
50%         NaN       975.26
75%         NaN     1,606.12
max         NaN     7,985.55
```

2.2. Dades faltants per columna

[31]:
```
missing_values_table(df_filtered)
```

```
El DataFrame seleccionat conté 17 columnes.
Hi han 4 columnes amb valors faltants.
```

[31]:

	Valors Faltants	% de Valors Totals
ActualElapsedTime	8387	0.40
AirTime	8387	0.40
ArrDelay	8387	0.40
ArrTime	7110	0.40

[32]:
```
df_filtered.dropna(axis=0, inplace=True)
```

[33]:
```
missing_values_table(df_filtered)
```

```
El DataFrame seleccionat conté 17 columnes.
Hi han 0 columnes amb valors faltants.
```

2.3. Creació de columnes noves Creem les columnes següents: - **IsWeekend**: Si el vol s'ha efectuat al cap de setmana o no - **Delayed** : Si el vol es va retrasar o no - **AvgSpeed** : Velocitat mitjana de vol (en km/h) - **TotalDelay**: Temps total de retràs (en minuts)

[34]:
```
df_filtered["IsWeekend"] = df_filtered["DayOfWeek"] >= 5
```

[35]:
```
df_filtered = df_filtered.assign(AvgSpeed = lambda x: x.Distance / x.AirTime)
```

[36]:
```
df_filtered = df_filtered.assign(Delayed = lambda x: (x.ArrDelay >= 15) | (x.
↪DepDelay >= 15))
```

[37]:
```
df_filtered["TotalDelay"] = df_filtered["ArrDelay"] + df_filtered["DepDelay"]
```

[38]:
```
df_filtered
```

[38]:

	Year	Month	DayofMonth	DayOfWeek	FlightNum	DepTime	CRSDepTime	\
0	2008	1	3	4	335	2,003.00	1955	
1	2008	1	3	4	3231	754.00	735	
2	2008	1	3	4	448	628.00	620	
4	2008	1	3	4	3920	1,829.00	1755	
5	2008	1	3	4	378	1,940.00	1915	
…	…	…	…	…	…	…	…	
7009710	2008	12	13	6	1621	1,250.00	1220	
7009717	2008	12	13	6	1631	657.00	600	
7009718	2008	12	13	6	1631	1,007.00	847	
7009726	2008	12	13	6	1639	1,251.00	1240	
7009727	2008	12	13	6	1641	1,110.00	1103	

	ArrTime	CRSArrTime	UniqueCarrier	ActualElapsedTime	AirTime	\
0	2,211.00	2225	WN	128.00	1.93	
1	1,002.00	1000	WN	128.00	1.88	
2	804.00	750	WN	96.00	1.27	
4	1,959.00	1925	WN	90.00	1.28	
5	2,121.00	2110	WN	101.00	1.45	
…	…	…	…	…	…	
7009710	1,617.00	1552	DL	147.00	2.00	
7009717	904.00	749	DL	127.00	1.30	
7009718	1,149.00	1010	DL	162.00	2.03	
7009726	1,446.00	1437	DL	115.00	1.48	
7009727	1,413.00	1418	DL	123.00	1.73	

	ArrDelay	DepDelay	Origin	Dest	Distance	IsWeekend	AvgSpeed	\
0	-14.00	8.00	IAD	TPA	1,303.57	False	674.26	
1	2.00	19.00	IAD	TPA	1,303.57	False	692.16	
2	14.00	8.00	IND	BWI	828.81	False	654.32	
4	34.00	34.00	IND	BWI	828.81	False	645.83	
5	11.00	25.00	IND	JAX	1,107.23	False	763.60	
…	…	…	…	…	…	…	…	

```
7009710     25.00     30.00    MSP  ATL  1,458.06       True    729.03
7009717     75.00     57.00    RIC  ATL    774.09       True    595.46
7009718     99.00     80.00    ATL  IAH  1,108.84       True    545.33
7009726      9.00     11.00    IAD  ATL    857.78       True    578.28
7009727     -5.00      7.00    SAT  ATL  1,406.56       True    811.48

         Delayed  TotalDelay
0          False       -6.00
1           True       21.00
2          False       22.00
4           True       68.00
5           True       36.00
…           …          …
7009710     True       55.00
7009717     True      132.00
7009718     True      179.00
7009726    False       20.00
7009727    False        2.00

[1928371 rows x 21 columns]
```

2.4. Aerolínies amb més endarreriements acumulats

[39]:
```
carriers = pd.read_csv("airlines.csv")
carriers = carriers.rename(columns={"Code":"UniqueCarrier", "Description":␣
↪"CarrierName"})
carriers.head()
```

[39]:
```
  UniqueCarrier                    CarrierName
0           02Q                 Titan Airways
1           04Q             Tradewind Aviation
2           05Q            Comlux Aviation, AG
3           06Q  Master Top Linhas Aereas Ltd.
4           07Q            Flair Airlines Ltd.
```

[40]:
```
df_filtered = pd.merge(df_filtered, carriers, how="left", on="UniqueCarrier")
df_filtered
```

[40]:
```
         Year  Month  DayofMonth  DayOfWeek  FlightNum  DepTime  CRSDepTime  \
0        2008      1           3          4        335 2,003.00        1955
1        2008      1           3          4       3231   754.00         735
2        2008      1           3          4        448   628.00         620
3        2008      1           3          4       3920 1,829.00        1755
4        2008      1           3          4        378 1,940.00        1915
…         …    …    …          …          …          …       …          …
1928366  2008     12          13          6       1621 1,250.00        1220
1928367  2008     12          13          6       1631   657.00         600
1928368  2008     12          13          6       1631 1,007.00         847
```

```
1928369  2008     12          13          6       1639 1,251.00        1240
1928370  2008     12          13          6       1641 1,110.00        1103

         ArrTime  CRSArrTime UniqueCarrier  ActualElapsedTime  AirTime  \
0       2,211.00        2225            WN             128.00     1.93
1       1,002.00        1000            WN             128.00     1.88
2         804.00         750            WN              96.00     1.27
3       1,959.00        1925            WN              90.00     1.28
4       2,121.00        2110            WN             101.00     1.45
…            …           …             …                 …        …
1928366 1,617.00        1552            DL             147.00     2.00
1928367   904.00         749            DL             127.00     1.30
1928368 1,149.00        1010            DL             162.00     2.03
1928369 1,446.00        1437            DL             115.00     1.48
1928370 1,413.00        1418            DL             123.00     1.73

         ArrDelay  DepDelay Origin Dest  Distance  IsWeekend  AvgSpeed  \
0          -14.00      8.00    IAD  TPA  1,303.57      False    674.26
1            2.00     19.00    IAD  TPA  1,303.57      False    692.16
2           14.00      8.00    IND  BWI    828.81      False    654.32
3           34.00     34.00    IND  BWI    828.81      False    645.83
4           11.00     25.00    IND  JAX  1,107.23      False    763.60
…             …        …    …   …  …       …         …        …
1928366     25.00     30.00    MSP  ATL  1,458.06       True    729.03
1928367     75.00     57.00    RIC  ATL    774.09       True    595.46
1928368     99.00     80.00    ATL  IAH  1,108.84       True    545.33
1928369      9.00     11.00    IAD  ATL    857.78       True    578.28
1928370     -5.00      7.00    SAT  ATL  1,406.56       True    811.48

         Delayed  TotalDelay            CarrierName
0          False       -6.00  Southwest Airlines Co.
1           True       21.00  Southwest Airlines Co.
2          False       22.00  Southwest Airlines Co.
3           True       68.00  Southwest Airlines Co.
4           True       36.00  Southwest Airlines Co.
…            …          …                    …
1928366     True       55.00    Delta Air Lines Inc.
1928367     True      132.00    Delta Air Lines Inc.
1928368     True      179.00    Delta Air Lines Inc.
1928369    False       20.00    Delta Air Lines Inc.
1928370    False        2.00    Delta Air Lines Inc.

[1928371 rows x 22 columns]
```

[41]:
```
carriers_delays = df_filtered[df_filtered["Delayed"] == True]
carriers_delays = carriers_delays[["UniqueCarrier", "CarrierName"]]
carriers_delays.value_counts()
```

[41]:
```
UniqueCarrier  CarrierName
WN             Southwest Airlines Co.          252227
AA             American Airlines Inc.          152051
UA             United Air Lines Inc.           111338
MQ             Envoy Air                       110760
OO             SkyWest Airlines Inc.            99705
DL             Delta Air Lines Inc.             84843
XE             ExpressJet Airlines Inc. (1)     81574
CO             Continental Air Lines Inc.       72669
US             US Airways Inc.                  71366
EV             ExpressJet Airlines Inc.         64646
NW             Northwest Airlines Inc.          61508
YV             Mesa Airlines Inc.               56100
FL             AirTran Airways Corporation      53517
OH             Comair Inc.                      45281
B6             JetBlue Airways                  44430
9E             Endeavor Air Inc.                40739
AS             Alaska Airlines Inc.             28329
F9             Frontier Airlines Inc.           18846
HA             Hawaiian Airlines Inc.            4705
dtype: int64
```

2.5. Vols més llargs

[46]:
```
longest_flights = df_filtered[["FlightNum", "AirTime"]]
longest_flights = longest_flights.sort_values(by="AirTime", ascending=False)
longest_flights.head(10) #AirTime in hours
```

[46]:

	FlightNum	AirTime
1483013	21	18.18
1361802	28	12.22
361059	15	11.07
554216	15	10.92
554220	15	10.90
554214	15	10.90
554212	15	10.87
554221	15	10.85
361381	15	10.82
362273	15	10.80

2.6. Vols més endarrerits

[48]:
```
delayed_flights = df_filtered[["FlightNum", "ArrDelay", "DepDelay",␣
↪"TotalDelay"]]
delayed_flights = delayed_flights.sort_values(by="TotalDelay", ascending=False)
delayed_flights.head(10)
```

[48]:

	FlightNum	ArrDelay	DepDelay	TotalDelay
683459	1699	2,453.00	2,467.00	4,920.00

```
321234          808  2,461.00  2,457.00    4,918.00
836341         1107  1,951.00  1,952.00    3,903.00
1005904        3538  1,707.00  1,710.00    3,417.00
1873807         357  1,655.00  1,597.00    3,252.00
1492136         512  1,583.00  1,552.00    3,135.00
682882         1472  1,542.00  1,545.00    3,087.00
1210247         804  1,510.00  1,518.00    3,028.00
519080         1743  1,490.00  1,490.00    2,980.00
542931         2093  1,370.00  1,521.00    2,891.00
```

2.7. Vols amb major distància recorreguda

[47]:
```
most_distant_flights = df_filtered[["Origin", "Dest", "Distance"]].
 ↪sort_values(["Distance"], ascending=False)
most_distant_flights = most_distant_flights.groupby(["Distance", "Origin"],␣
 ↪sort=False)
most_distant_flights.first().head(10)
```

[47]:
```
                Dest
Distance Origin
7,985.55 EWR     HNL
         HNL     EWR
7,245.25 ATL     HNL
         HNL     ATL
6,828.43 ORD     HNL
         HNL     ORD
6,780.15 KOA     ORD
6,733.48 ORD     OGG
6,392.30 MSP     HNL
         HNL     MSP
```

3.0. Exportació del dataset a Excel

[49]:
```
DelayedFlightsFiltered = df_filtered.to_csv("DelayedFlightsFiltered.csv")
```