

```
In [2]: #Explora les dades que conté, i queda't únicament amb les columnes que consideris rellevants
```

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
```

```

# Import the dataset using read_csv()
df = pd.read_csv("delayedFlights.csv")

# Print out the first 10 rows of the dataset
df

Out[2]:
   Unnamed: 0  Year  Month  DayOfMonth  DayOfWeek  DepTime  CRSDepTime  ArrTime  CRSArrTime  UniqueCarrier  ...  TaxiIn  TaxiOut  Cancelled  CancellationCode  Diverted  CarrierDelay  WeatherDelay
0            0    2009     1         3           4    2093.0      1955     2211.0      2225         WN          ...      5.0      8.0          0           N          0         NaN
1            1    2009     1         3           4    754.0       735     1052.0      1000         WN          ...      4.0      9.0          0           N          0         NaN
2            2    2009     1         3           4    628.0       620     804.0       750         WN          ...      3.0     17.0          0           N          0         NaN
3            3    4    2009     1         3           4    1829.0     1755     1959.0     1825         WN          ...      3.0     10.0          0           N          0         2.0
4            4    5    2009     1         3           4    1940.0     1915     2121.0     2110         WN          ...      4.0     10.0          0           N          0         NaN
...         ...    ...    ...         ...         ...         ...         ...         ...         ...         ...          ...      ...      ...          ...           ...          ...         ...
1936753      7009710  2008     12        13           6    1250.0      1220     1617.0     1552         DL          ...      9.0     18.0          0           N          0         3.0
1936754      7009717  2008     12        13           6     657.0       600     904.0       749         DL          ...     15.0     34.0          0           N          0         0.0
1936755      7009718  2008     12        13           6    1067.0       847     1149.0     1010         DL          ...      8.0     32.0          0           N          0         1.0
1936756      7009726  2008     12        13           6    1251.0      1240     1446.0     1437         DL          ...     13.0     13.0          0           N          0         NaN
1936757      7009727  2008     12        13           6    1110.0      1103     1413.0     1418         DL          ...      8.0     11.0          0           N          0         NaN

1936758 rows x 30 columns

```

0	0	2008	1	3	4	2003.0	1955	2211.0	2225	WN ...	4.0	8.0	0	N	0	NaN	N
1	1	2008	1	3	4	754.0	735	1002.0	1000	WN ...	5.0	10.0	0	N	0	NaN	N

```
In [4]: subset = df[['Year', 'Month', 'DayOfMonth', 'DayOfWeek', 'ArrTime', 'FlightNum', 'Cancelled', 'DepTime']]
subset = subset.drop(range(11, 1936758), axis=0)
print(subset)
```

	Year	Month	DayOfMonth	DayOfWeek	ArrTime	FlightNum	Cancelled	DepTime
0	2008	1	3	4	2211.0	335	0	2083.0
1	2008	1	3	4	1862.0	3231	0	754.0
2	2008	1	3	4	894.0	648	0	628.0
3	2008	1	3	4	1959.0	3920	0	1829.0
4	2008	1	3	4	2121.0	378	0	1946.0
5	2008	1	3	4	2037.0	599	0	1937.0
6	2008	1	3	4	916.0	100	0	766.0
7	2008	1	3	4	1645.0	1133	0	1644.0
8	2008	1	3	4	1021.0	2272	0	1029.0

```

9 2008 1 3 4 1640.0 675 0 1452.0
10 2008 1 3 4 940.0 1144 0 754.0

```

In [5]: `#es un informe complet del data set.`

```

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

def func_2(column = 'Year'):
    year= df[column].describe()
    print(year)

func_2('Month')
func_2('DayOfMonth')
func_2('DayOfWeek')
func_2('ArrTime')
```

```
func_2['FlightNum')
func_2['Cancelled']
# jirreard flight-no tot en un dataframe

count      1.936758e+06
mean        6.111186e+00
std          3.482546e+00
min          1.000000e+00
25%          3.000000e+00
50%          6.000000e+00
75%          9.000000e+00
max          1.200000e+01
Name: Month, dtype: float64
count      1.936758e+06
mean        1.575347e+01
std          8.776272e+00
min          1.000000e+00
25%          8.000000e+00
50%          1.600000e+01
75%          2.300000e+01
max          3.100000e+01
Name: DayOfMonth, dtype: float64
count      1.936758e+06
mean        3.984827e+00
std          1.995064e+00
min          1.000000e+00
25%          2.000000e+00
50%          4.000000e+00
75%          6.000000e+00
max          7.000000e+00
Name: DayOfWeek, dtype: float64
count      1.929648e+06
mean        1.630643e+03
std          5.481781e+02
min          1.000000e+00
25%          1.310000e+03
50%          1.715000e+03
75%          2.630000e+03
max          2.400000e+03
Name: ArrTime, dtype: float64
count      1.936758e+06
mean        2.184263e+03
std          1.944702e+03
min          1.000000e+00
25%          6.100000e+02
50%          1.543000e+03
75%          3.422000e+03
max          9.742000e+03
Name: FlightNum, dtype: float64
count      1.936758e+06
mean        3.269348e-04
std          1.007562e-02
min          0.000000e+00
25%          0.000000e+00
50%          0.000000e+00
75%          0.000000e+00
max          1.000000e+00
Name: Cancelled, dtype: float64
```

```
columna_1 = df[column]
res = columna_1.mean()
x = columna_1.fillna(res, inplace=True)
print(x)

function()
Function('Month')
Function('DayOfMonth')
Function('DayOfWeek')
Function('ArrTime')
Function('FlightNum')
Function('Cancelled')
Function('CarrierDelay')

None
None
None
None
None
None
None
None
```

In [7]: `#Crea columnes noves (velocitat mitjana del vol, si ha arribat tard o no...)`

```
subset.insert(4, 'AvSpeed', ['389', '435', '521', '438', '420', '387', '554', '398', '465', '654', '567'], True )
```

```
subset.insert(6, "Delayed", [Yes, 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No'], True)
subset.insert(7, "Passengers", ['350', '560', '480', '650', '380', '430', '680', '420', '550', '540', '370', '520'], True)
print(subset)
```

	Year	Month	DayOfMonth	DayOfWeek	AVSpeed	ArrTime	Delayed	Passengers	\
0	2008	1	3	4	389	2211.0	Yes	350	
1	2008	1	3	4	435	1802.0	No	560	
2	2008	1	3	4	521	884.0	No	480	
3	2008	1	3	4	438	1959.0	No	650	
4	2008	1	3	4	420	2121.0	No	380	
5	2008	1	3	4	387	2037.0	Yes	430	
6	2008	1	3	4	554	915.0	No	420	
7	2008	1	3	4	398	1845.0	Yes	550	
8	2008	1	3	4	465	1021.0	No	540	
9	2008	1	3	4	654	1640.0	No	370	
10	2008	1	3	4	567	940.0	No	520	

```
FlightNum Canceled Deptime
```

0	335	0	2893.0
1	3233	0	754.0
2	448	0	628.0
3	3920	0	1829.0
4	378	0	1940.0
5	509	0	1937.0
6	188	0	786.0
7	1533	0	1844.0
8	2272	0	1829.0
9	675	0	1452.0
10	1144	0	754.0

```
In [8]: #Taula de les aerolínies amb més endarrerriments acumulats

newDet = df[['CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay']]

ordenarTard = newDet.sort_values(['CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay'])

newDet = ordenarTard.sum(axis=1)
print(newDet.tail())

685437    1542.0
1497823    1563.0
1089553    1767.0
839396     1951.0
686014     2453.0
dtype: float64
```

```
##Quins són els vols més llargs? I els més endarrerits?

resta = df['ArrTime'] - df['DepTime']
print(resta)

x = resta.sort_values()
print(x)

x.tail(20)
```

0	208.0
1	248.0
2	176.0
3	130.0
4	181.0
	...
1936753	307.0
1936754	247.0
1936755	142.0
1936756	490.0

```
1936757 303.0
1936757 303.0
Length: 1936758, dtype: float64
1262416 -2396.0
1178372 -2394.0
1087824 -2391.0
284445 -2385.0
1352978 -2379.0
...
1176923 2357.0
1260384 2377.0
1656812 2385.0
1179896 2392.0
1765176 2397.0
Length: 1936758, dtype: float64
184434 1609.140629
287298 2387.000000
1352198 2339.000000
932288 2340.000000
1691340 2342.000000
1352192 2342.000000
```

```
1691849      2342.000000
675390       2343.000000
987881       2345.000000
472955       2347.000000
726753       2347.000000
280274       2348.000000
1179548      2348.000000
1175548      2351.000000
1793034      2354.000000
1176623      2357.000000
1260284      2377.000000
1656812      2385.000000
1179896      2392.000000
1765176      2397.000000
dtype: float64
```

```
In [57]: subset.to_excel('novaTaula.xlsx')
```

Unique Carrier

ArDelay	Count
-15	33000
-14	0
-13	0
-12	0
-11	0
-10	0
-9	0
-8	0
-7	0
-6	0
-5	0
-4	0
-3	0
-2	0
-1	0
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	0
36	0
37	0
38	0
39	0
40	0
41	0
42	0
43	0
44	0
45	0
46	0
47	0
48	0
49	0
50	0
51	0
52	0
53	0
54	0
55	0
56	0
57	0
58	0
59	0
60	0
61	0
62	0
63	0
64	0
65	0
66	0
67	0
68	0
69	0
70	0
71	0
72	0
73	0
74	0
75	0
76	0
77	0
78	0
79	0
80	0
81	0
82	0
83	0
84	0
85	0
86	0
87	0
88	0
89	0
90	0
91	0
92	0
93	0
94	0
95	0
96	0
97	0
98	0
99	0
100	0

Country	Number of Publications
VN	22000
AA	19000
MO	14000
UA	14000
CO	13000
DL	11000
XE	10000
OS	10000
US	9500
EV	8000
TW	7500
FL	7000
YV	6500
BB	5500
CH	5000
BE	5000
AS	4000
FR	3000
HA	1000
AQ	500

Carrier	Percentage
WN	10.50
AS	2.03
9E	2.02
OH	2.02
B6	2.02
YV	2.02
FL	2.02
NW	2.02
EV	2.02
US	2.02
CO	2.02
IE	2.02
DL	2.02
OO	2.02
UA	2.02
9A	2.02

The graph displays two data series over five time segments. The blue series has discrete data points, while the olive series is continuous.

Segment	Blue Line (Discrete)	Olive Line (Continuous)
1	Low	Low, then rises to mid
2	Mid-High	Mid, then rises to High
3	High	High, then falls to Low
4	Mid	Low, then rises to Mid
5	High	Mid, then falls to Low

The graph displays two data series over time (t). The x-axis ranges from 0 to 40,000 with major ticks every 10,000. The y-axis, labeled 'VN', ranges from 0 to 10 with major ticks every 2 units. A dashed horizontal line is positioned at VN = 5. The blue line with circular markers shows high variability, with values ranging from approximately 2 to 10. The green line shows lower variability, with values ranging from approximately 6 to 10.

Dues variables different tipus

ArDelay

UniqueCarrier

Airline	Arr Delay (approx.)
Southwest	36,000
Delta	35,500
JetBlue	35,200
Allegiant	35,000
Frontier	34,800
Allegiant	34,500
Delta	34,200
Southwest	33,800
Delta	33,500
Allegiant	33,200
Delta	32,800
Allegiant	32,500
Delta	32,200
Allegiant	31,800
Delta	31,500
Allegiant	31,200
Delta	30,800
Allegiant	30,500
Delta	30,200
Allegiant	29,800
Delta	29,500
Allegiant	29,200
Delta	28,800

The top plot is a bar chart showing the distribution of 'UniqueCarrier' values. The x-axis represents 'UniqueCarrier' with values from 1.0 to 10.0. The y-axis represents frequency, ranging from 0 to 10000. The bars are colored in a light blue/grey shade.

The bottom plot is a scatter plot titled 'UniqueCarrier = WN'. The x-axis represents 'UniqueCarrier' with values from 1.0 to 10.0. The y-axis represents 'AirTime', ranging from 100 to 120. The plot shows a single data point for 'UniqueCarrier' = 1.0, which is colored green. A legend on the right indicates the color mapping for 'AirTime' values: 37.0 (red), 47.0 (orange), and 49.0 (green).

A scatter plot showing the relationship between ArtDaisy (x-axis) and DevDaisy (y-axis) for the 'WN' carrier. The data points are blue dots, and a solid blue regression line is shown with a light blue shaded confidence interval. The axes range from 0 to 80. The plot shows a positive correlation, with the regression line passing through the center of the data points.