Data Science amb Python

Estudiant: Cristiane de Souza da Silva

Lliurament tasca 5: Exploració de les dades

Descripció

Familiaritza't amb les tecniques de exploració de les dades mitjantçant la estructure de dades, Datafreame amb la llibreria Pandas.

This dataset is composed by the following variables:

1) Year 2008

2) Month 1-12

3) DayofMonth 1-31

4) DayOfWeek 1 (Monday) - 7 (Sunday)

5) DepTime actual departure time (local, hhmm)

6) CRSDepTime scheduled departure time (local, hhmm)

7) ArrTime actual arrival time (local, hhmm)

8) CRSArrTime scheduled arrival time (local, hhmm)

9) UniqueCarrier unique carrier code

10) FlightNum flight number

11) TailNum plane tail number: aircraft registration, unique aircraft identifier

12) ActualElapsedTime in minutes

13) CRSElapsedTime in minutes

14) AirTime in minutes

15) ArrDelay arrival delay, in minutes: 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).

16) DepDelay departure delay, in minutes

17) Origin origin IATA airport code

18) Dest destination IATA airport code

19) Distance in miles

20) TaxiIn taxi in time, in minutes

21) TaxiOut taxi out time in minutes

22) Cancelled *was the flight cancelled

23) CancellationCode reason for cancellation (A = carrier, B = weather, C = NAS, D = security)

24) Diverted 1 = yes, 0 = no

25) CarrierDelay in minutes: 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.

26) WeatherDelay in minutes: 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.

27) NASDelay in minutes: 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.

28) SecurityDelay in minutes: 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.

29) LateAircraftDelay in minutes: 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.

- **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 [1]:
```
# Loading the libraries

import numpy as np
import pandas as pd


flights = pd.read_csv('DelayedFlights.csv')
flights
```

Out[1]:

	Unnamed: 0	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTi
0	0	2008	1	3	4	2003.0	1955	221
1	1	2008	1	3	4	754.0	735	100
2	2	2008	1	3	4	628.0	620	80
3	4	2008	1	3	4	1829.0	1755	195
4	5	2008	1	3	4	1940.0	1915	212
...	...	...	...	...	...	...	...	
1936753	7009710	2008	12	13	6	1250.0	1220	161
1936754	7009717	2008	12	13	6	657.0	600	90
1936755	7009718	2008	12	13	6	1007.0	847	114
1936756	7009726	2008	12	13	6	1251.0	1240	144
1936757	7009727	2008	12	13	6	1110.0	1103	141

1936758 rows × 30 columns

In [2]:
```
flights.columns
```

Out[2]:
```
Index(['Unnamed: 0', 'Year', 'Month', 'DayofMonth', 'DayOfWeek', 'DepTime',
       'CRSDepTime', 'ArrTime', 'CRSArrTime', 'UniqueCarrier', 'FlightNum',
       'TailNum', 'ActualElapsedTime', 'CRSElapsedTime', 'AirTime', 'ArrDel
ay',
       'DepDelay', 'Origin', 'Dest', 'Distance', 'TaxiIn', 'TaxiOut',
       'Cancelled', 'CancellationCode', 'Diverted', 'CarrierDelay',
       'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay'],
      dtype='object')
```

In [3]:
```
# Remove 'Unnamed' column

flights.drop('Unnamed: 0', axis=1, inplace=True)
```

In [4]:
```
print(flights.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1936758 entries, 0 to 1936757
Data columns (total 29 columns):
 #   Column             Dtype  
---  ------             -----  
 0   Year               int64  
 1   Month              int64  
 2   DayofMonth         int64  
 3   DayOfWeek          int64  
 4   DepTime            float64
 5   CRSDepTime         int64  
 6   ArrTime            float64
 7   CRSArrTime         int64  
 8   UniqueCarrier      object 
 9   FlightNum          int64  
 10  TailNum            object 
 11  ActualElapsedTime  float64
 12  CRSElapsedTime     float64
 13  AirTime            float64
 14  ArrDelay           float64
 15  DepDelay           float64
 16  Origin             object 
 17  Dest               object 
 18  Distance           int64  
 19  TaxiIn             float64
 20  TaxiOut            float64
 21  Cancelled          int64  
 22  CancellationCode   object 
 23  Diverted           int64  
 24  CarrierDelay       float64
 25  WeatherDelay       float64
 26  NASDelay           float64
 27  SecurityDelay      float64
 28  LateAircraftDelay  float64
dtypes: float64(14), int64(10), object(5)
memory usage: 428.5+ MB
None
```

- **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 [5]:
```
# Summarize the columns of interest statistically

flights.describe()
```

Out[5]:

	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTim
count	1936758.0	1.936758e+06	1.936758e+06	1.936758e+06	1.936758e+06	1.936758e+0
mean	2008.0	6.111106e+00	1.575347e+01	3.984827e+00	1.518534e+03	1.467473e+0
std	0.0	3.482546e+00	8.776272e+00	1.995966e+00	4.504853e+02	4.247668e+0
min	2008.0	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	0.000000e+0
25%	2008.0	3.000000e+00	8.000000e+00	2.000000e+00	1.203000e+03	1.135000e+0
50%	2008.0	6.000000e+00	1.600000e+01	4.000000e+00	1.545000e+03	1.510000e+0
75%	2008.0	9.000000e+00	2.300000e+01	6.000000e+00	1.900000e+03	1.815000e+0
max	2008.0	1.200000e+01	3.100000e+01	7.000000e+00	2.400000e+03	2.359000e+0

8 rows × 24 columns

In [6]:
```
# Find how many missing data are per column

flights.isna().sum()
```

Out[6]:
```
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
```

In [ ]:

In [7]:
```
#Create new columns
# Create departure date column
# Convert time

flights['DepDate'] = pd.to_datetime(flights.Year*10000+flights.Month*100+fl
```

In [8]:
```
flights['DepDate']
```

Out[8]:
```
0          2008-01-03
1          2008-01-03
2          2008-01-03
3          2008-01-03
4          2008-01-03
              ...
1936753    2008-12-13
1936754    2008-12-13
1936755    2008-12-13
1936756    2008-12-13
1936757    2008-12-13
Name: DepDate, Length: 1936758, dtype: datetime64[ns]
```

In [ ]:
```

```

- Exercici 3

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

In [9]:
```
# Remove the columns with missing data

flights.drop(['ArrTime','ActualElapsedTime','CRSElapsedTime','AirTime','Ar
flights
```

Out[9]:

	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	CRSArrTime	Uni
0	2008	1	3	4	2003.0	1955	2225	
1	2008	1	3	4	754.0	735	1000	
2	2008	1	3	4	628.0	620	750	
3	2008	1	3	4	1829.0	1755	1925	
4	2008	1	3	4	1940.0	1915	2110	
...	...	...	...	...	...	...	...	
1936753	2008	12	13	6	1250.0	1220	1552	
1936754	2008	12	13	6	657.0	600	749	
1936755	2008	12	13	6	1007.0	847	1010	
1936756	2008	12	13	6	1251.0	1240	1437	
1936757	2008	12	13	6	1110.0	1103	1418	

1936758 rows × 17 columns

In [11]:
```
#Export the data set clean and with the new columns to Excel.

flights.to_csv("output.csv")

# The file is too big to be exported to excel, so I exported as csv format
```

In [ ]:
```

```