LAB2

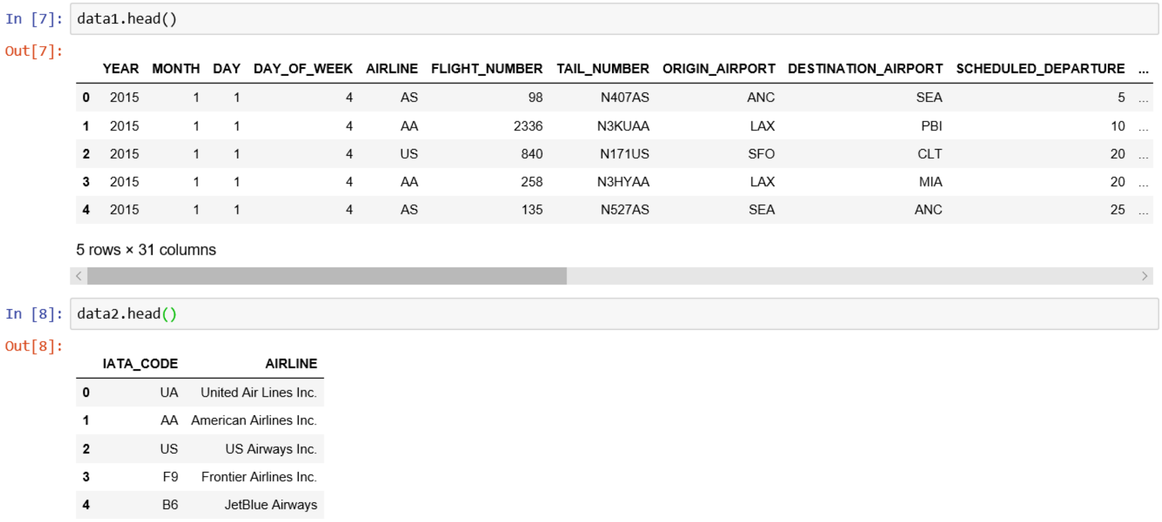
Chalakorn Manopirom

59070503409

Read two dataframe “flights.csv” and “airlines.csv”

➢ Hint: pd.read_csv

1. Print the first 5 rows of flight and airline dataframe



```
In [7]: data1.head()
Out[7]:
```

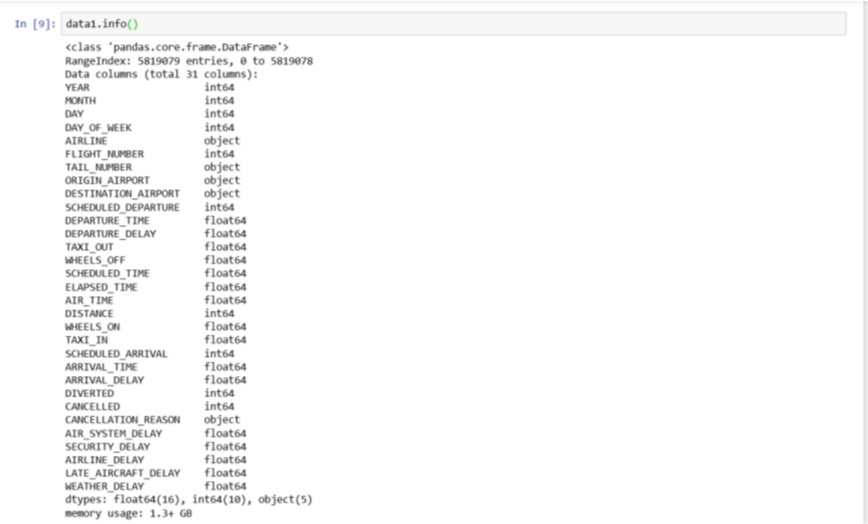
	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	...
0	2015	1	1	4	AS	98	N407AS	ANC	SEA	5	...
1	2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	10	...
2	2015	1	1	4	US	840	N171US	SFO	CLT	20	...
3	2015	1	1	4	AA	258	N3HYAA	LAX	MIA	20	...
4	2015	1	1	4	AS	135	N527AS	SEA	ANC	25	...

5 rows × 31 columns

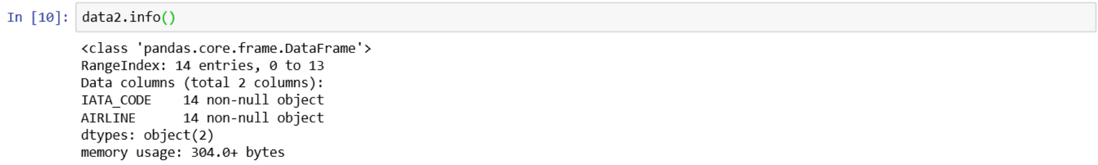
```
In [8]: data2.head()
Out[8]:
```

	IATA_CODE	AIRLINE
0	UA	United Air Lines Inc.
1	AA	American Airlines Inc.
2	US	US Airways Inc.
3	F9	Frontier Airlines Inc.
4	B6	JetBlue Airways

2. How many features in flight.csv and airline.csv

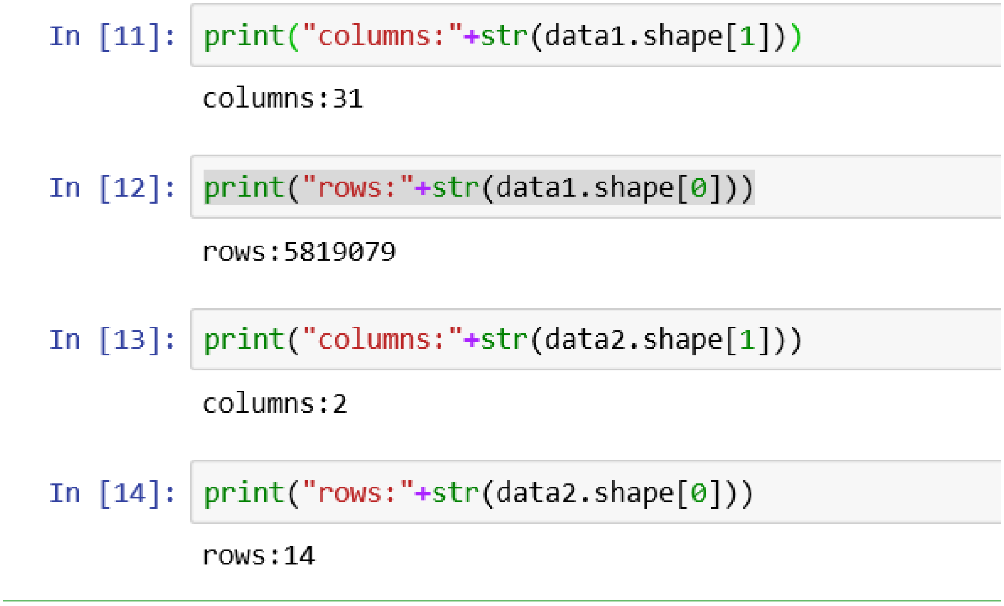


```
In [9]: data1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5819079 entries, 0 to 5819078
Data columns (total 31 columns):
YEAR                   int64
MONTH                  int64
DAY                    int64
DAY_OF_WEEK            int64
AIRLINE                object
FLIGHT_NUMBER          int64
TAIL_NUMBER            object
ORIGIN_AIRPORT         object
DESTINATION_AIRPORT    object
SCHEDULED_DEPARTURE    int64
DEPARTURE_TIME         float64
DEPARTURE_DELAY        float64
TAXI_OUT               float64
WHEELS_OFF             float64
SCHEDULED_TIME         float64
ELAPSED_TIME           float64
AIR_TIME               float64
DISTANCE               int64
WHEELS_ON              float64
TAXI_IN                float64
SCHEDULED_ARRIVAL      int64
ARRIVAL_TIME           float64
ARRIVAL_DELAY          float64
DIVERTED               int64
CANCELLED              int64
CANCELLATION_REASON    object
AIR_SYSTEM_DELAY       float64
SECURITY_DELAY         float64
AIRLINE_DELAY          float64
LATE_AIRCRAFT_DELAY    float64
WEATHER_DELAY          float64
dtypes: float64(16), int64(10), object(5)
memory usage: 1.3+ GB
```



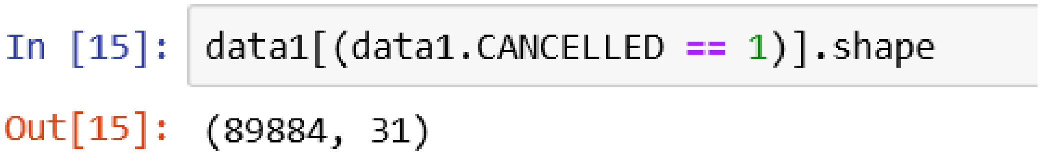
```
In [10]: data2.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 2 columns):
IATA_CODE     14 non-null object
AIRLINE       14 non-null object
dtypes: object(2)
memory usage: 304.0+ bytes
```

3. How many rows/records in flight.csv and airline.csv



```
In [11]: print("columns:"+str(data1.shape[1]))

columns:31

In [12]: print("rows:"+str(data1.shape[0]))

rows:5819079

In [13]: print("columns:"+str(data2.shape[1]))

columns:2

In [14]: print("rows:"+str(data2.shape[0]))

rows:14
```

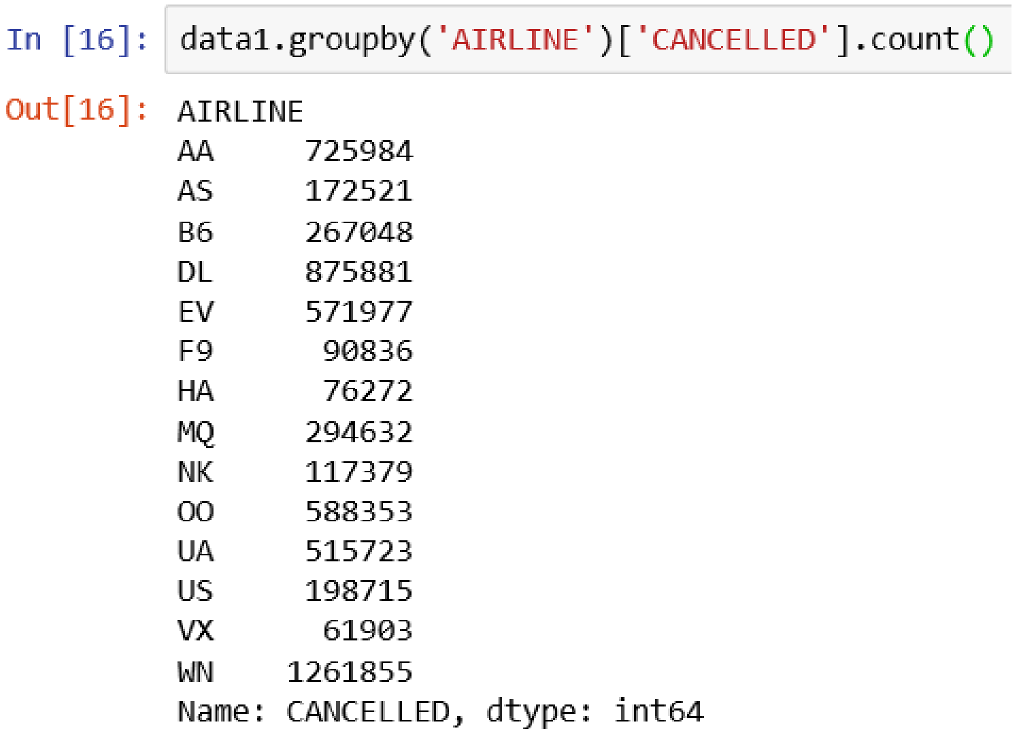
4. How many flight that was cancelled? [Hint : filter]



```
In [15]: data1[(data1.CANCELLED == 1)].shape

Out[15]: (89884, 31)
```

5. Which airline has the most cancelled flight?

[Hint: groupby]

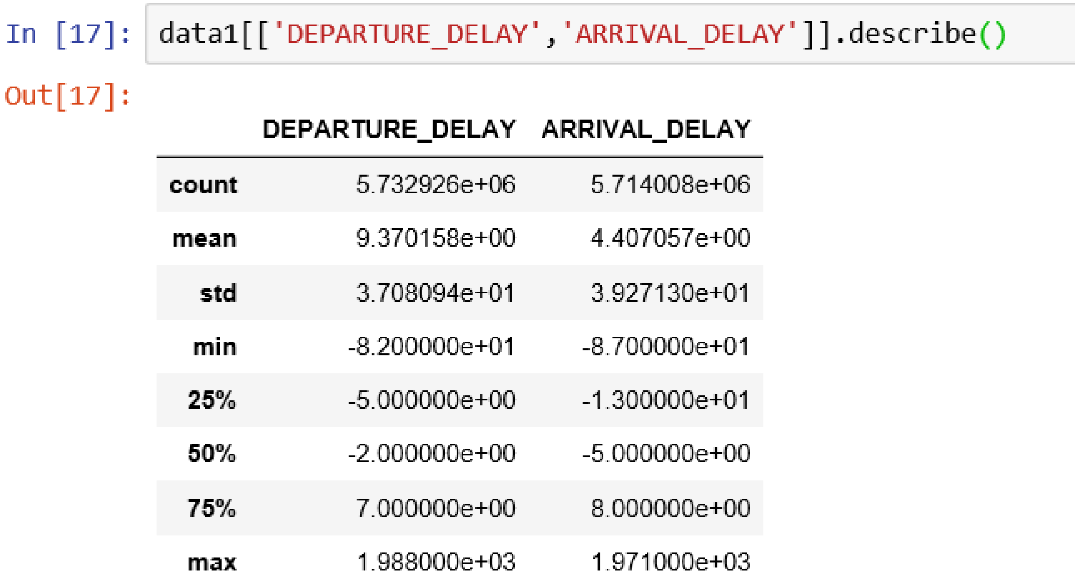


```
In [16]: data1.groupby('AIRLINE')['CANCELLED'].count()

Out[16]: AIRLINE
         AA     725984
         AS     172521
         B6     267048
         DL     875881
         EV     571977
         F9      90836
         HA      76272
         MQ     294632
         NK     117379
         OO     588353
         UA     515723
         US     198715
         VX      61903
         WN    1261855
         Name: CANCELLED, dtype: int64
```

6. What is the maximum, minimum, sd and mean of departure delay and arrival delay

Filter NA out of departure_delay and arrival_delay

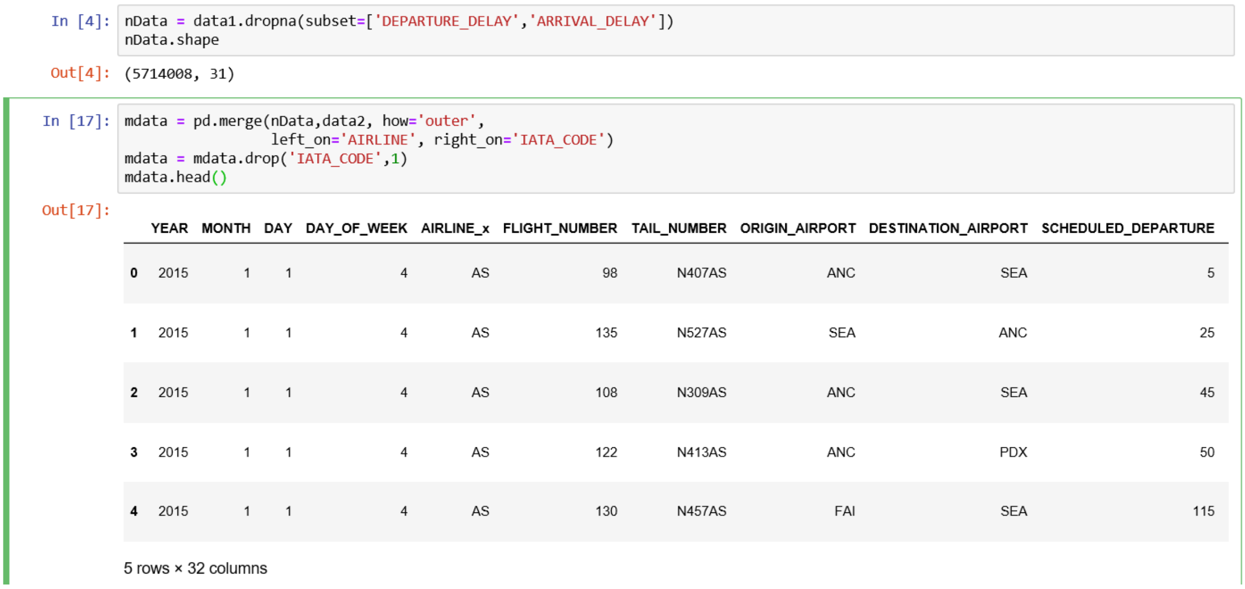
Join airlines data to flight data using ‘airline’ as a key



```
In [17]: data1[['DEPARTURE_DELAY','ARRIVAL_DELAY']].describe()

Out[17]:
```

	DEPARTURE_DELAY	ARRIVAL_DELAY
count	5.732926e+06	5.714008e+06
mean	9.370158e+00	4.407057e+00
std	3.708094e+01	3.927130e+01
min	-8.200000e+01	-8.700000e+01
25%	-5.000000e+00	-1.300000e+01
50%	-2.000000e+00	-5.000000e+00
75%	7.000000e+00	8.000000e+00
max	1.988000e+03	1.971000e+03

7.Print dataframe after join the data

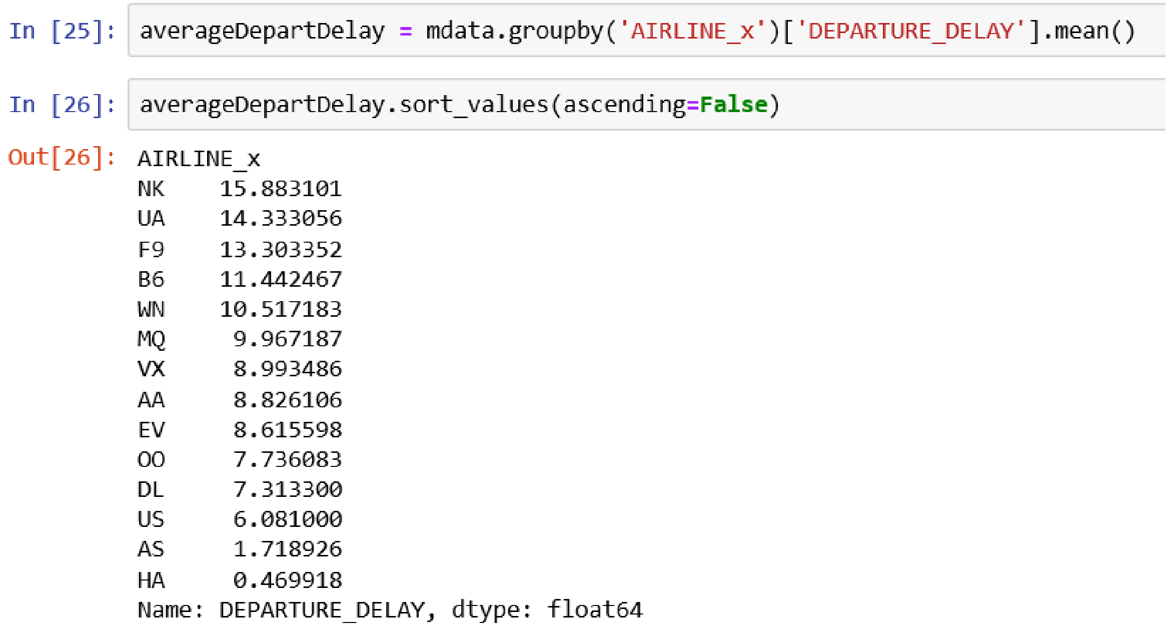


```
In [4]: nData = data1.dropna(subset=['DEPARTURE_DELAY','ARRIVAL_DELAY'])
        nData.shape

Out[4]: (5714008, 31)

In [17]: mdata = pd.merge(nData,data2, how='outer',
                          left_on='AIRLINE', right_on='IATA_CODE')
         mdata = mdata.drop('IATA_CODE',1)
         mdata.head()

Out[17]:
```

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_x	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE
0	2015	1	1	4	AS	98	N407AS	ANC	SEA	5
1	2015	1	1	4	AS	135	N527AS	SEA	ANC	25
2	2015	1	1	4	AS	108	N309AS	ANC	SEA	45
3	2015	1	1	4	AS	122	N413AS	ANC	PDX	50
4	2015	1	1	4	AS	130	N457AS	FAI	SEA	115

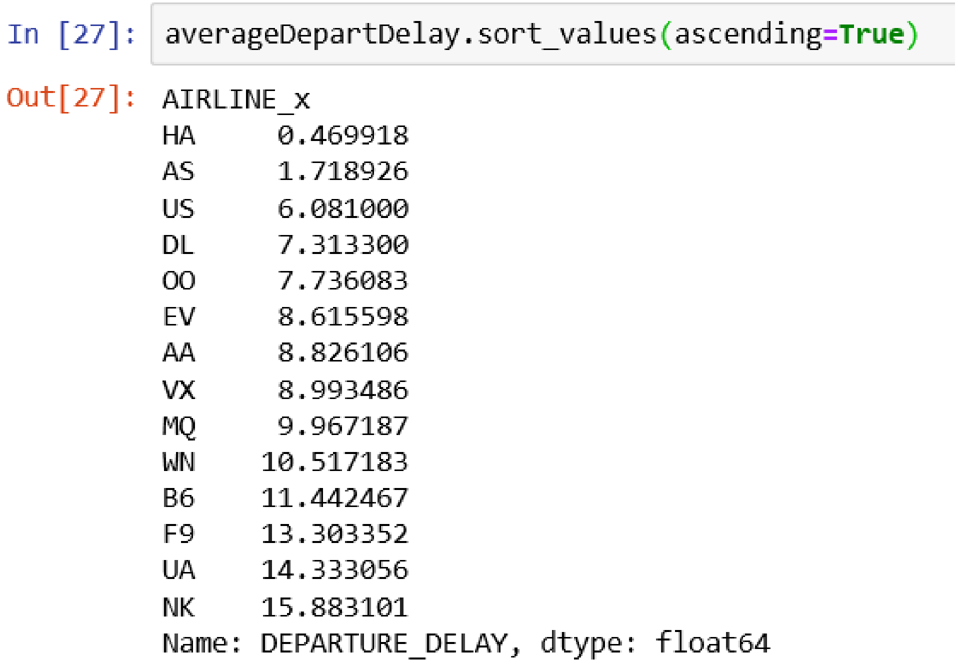
5 rows × 32 columns

8. Which airline has the highest average departure_delay time and how long ?



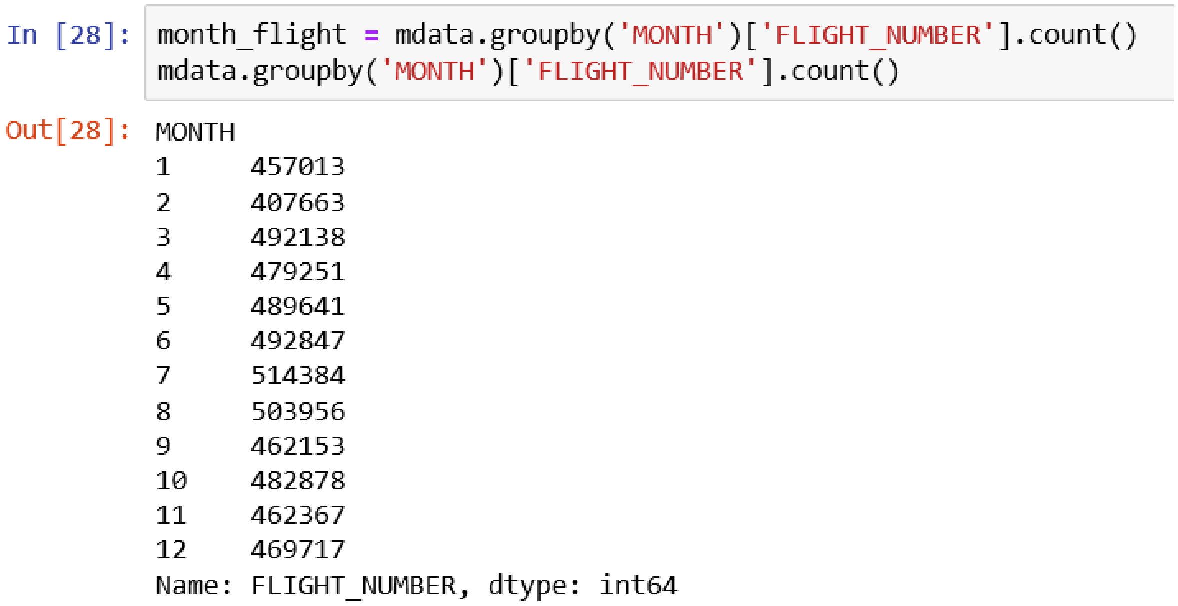
```
In [25]: averageDepartDelay = mdata.groupby('AIRLINE_x')['DEPARTURE_DELAY'].mean()

In [26]: averageDepartDelay.sort_values(ascending=False)

Out[26]: AIRLINE_x
         NK    15.883101
         UA    14.333056
         F9    13.303352
         B6    11.442467
         WN    10.517183
         MQ     9.967187
         VX     8.993486
         AA     8.826106
         EV     8.615598
         OO     7.736083
         DL     7.313300
         US     6.081000
         AS     1.718926
         HA     0.469918
         Name: DEPARTURE_DELAY, dtype: float64
```

9. Which airline has the lowest average departure_delay time and how long ?



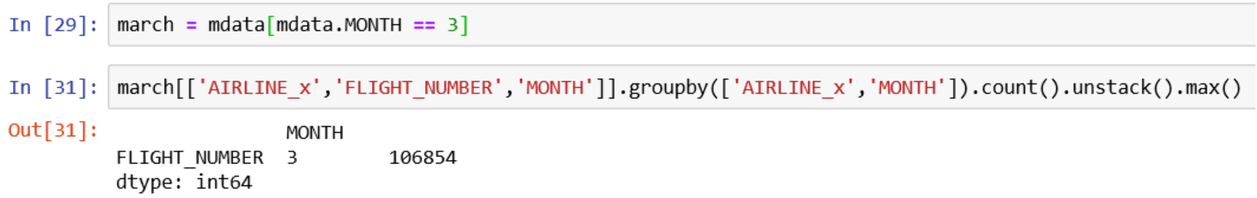
```
In [27]: averageDepartDelay.sort_values(ascending=True)

Out[27]: AIRLINE_x
         HA     0.469918
         AS     1.718926
         US     6.081000
         DL     7.313300
         OO     7.736083
         EV     8.615598
         AA     8.826106
         VX     8.993486
         MQ     9.967187
         WN    10.517183
         B6    11.442467
         F9    13.303352
         UA    14.333056
         NK    15.883101
         Name: DEPARTURE_DELAY, dtype: float64
```

10. Which month has the highest number of flight? and how many?



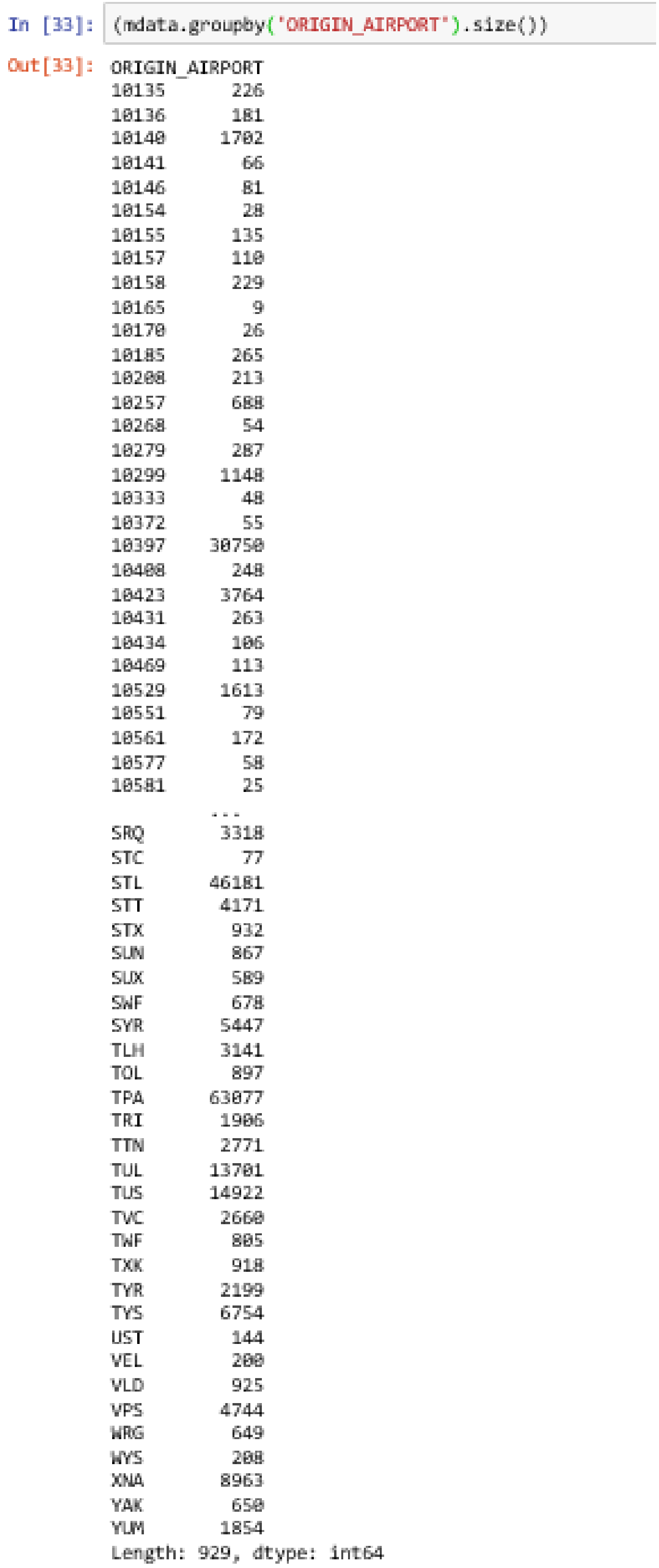
```
In [28]: month_flight = mdata.groupby('MONTH')['FLIGHT_NUMBER'].count()
         mdata.groupby('MONTH')['FLIGHT_NUMBER'].count()

Out[28]: MONTH
         1     457013
         2     407663
         3     492138
         4     479251
         5     489641
         6     492847
         7     514384
         8     503956
         9     462153
         10    482878
         11    462367
         12    469717
         Name: FLIGHT_NUMBER, dtype: int64
```

11. In March, which airline has the highest flight? And how many flights?



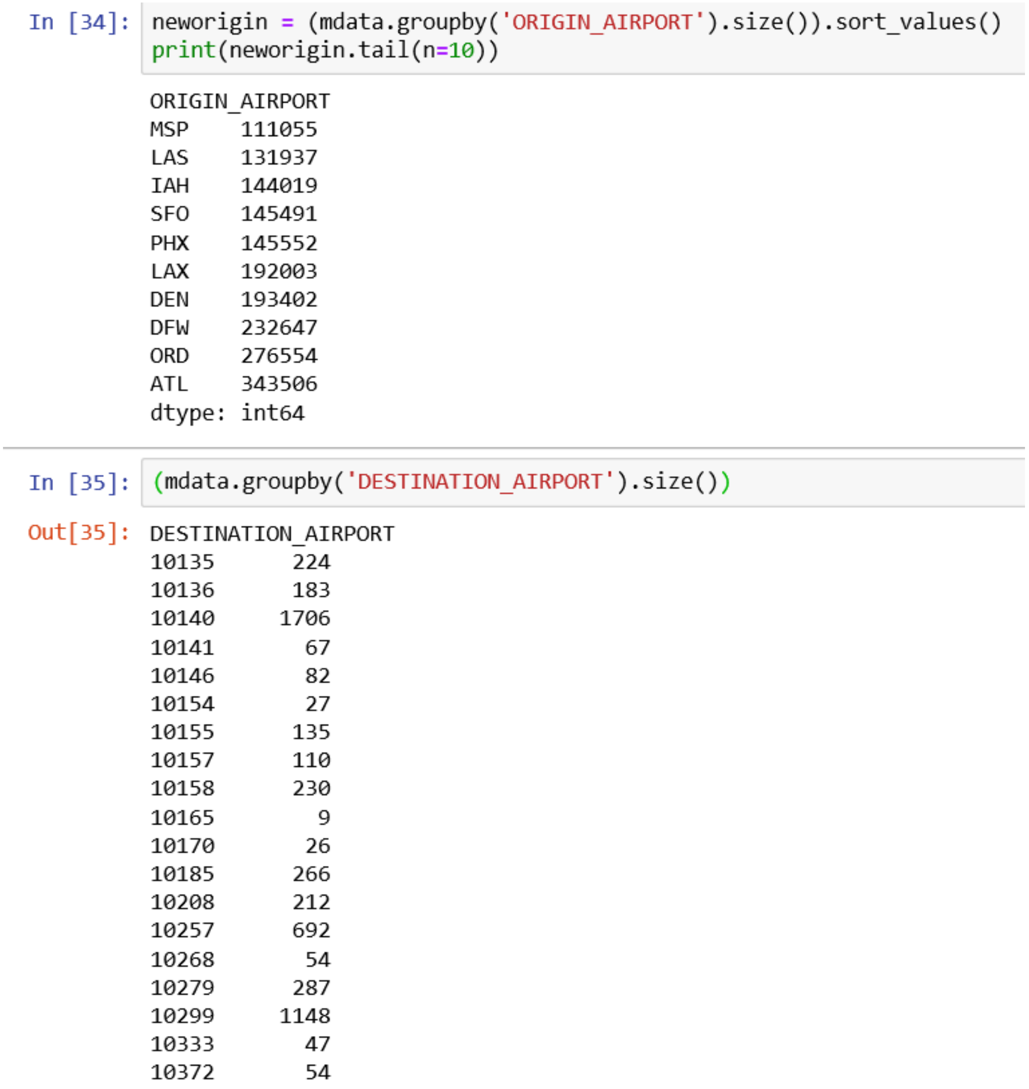
```
In [29]: march = mdata[mdata.MONTH == 3]

In [31]: march[['AIRLINE_x','FLIGHT_NUMBER','MONTH']].groupby(['AIRLINE_x','MONTH']).count().unstack().max()

Out[31]:                MONTH
         FLIGHT_NUMBER  3        106854
         dtype: int64
```

12. Which origin has the most rows? How many ? Which destination has the most rows? How many?



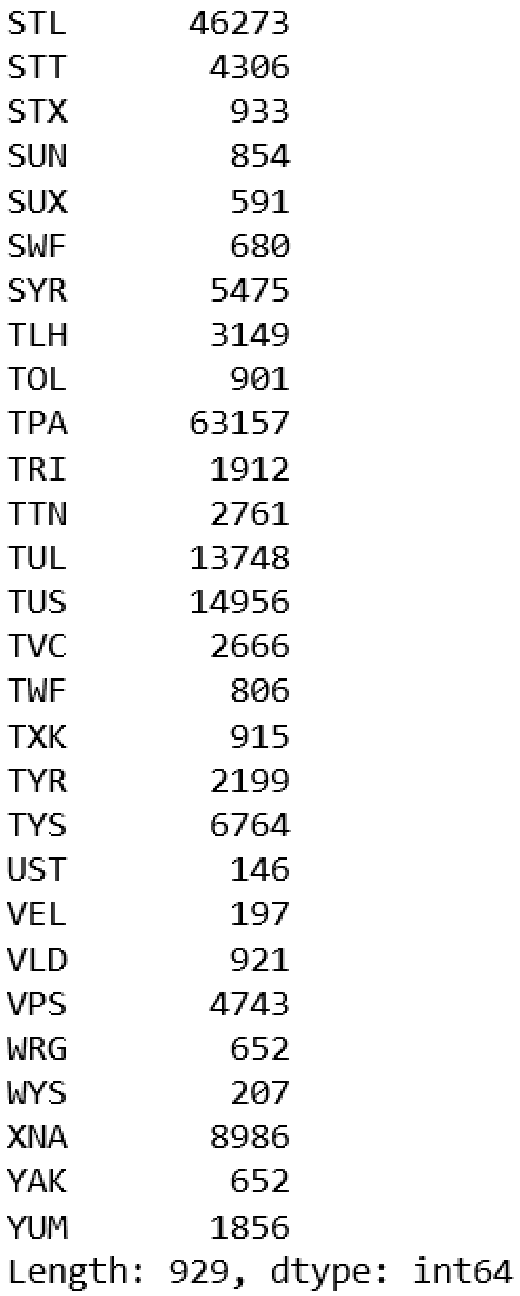
```
In [33]: (mdata.groupby('ORIGIN_AIRPORT').size())
```

```
Out[33]: ORIGIN_AIRPORT
         10135       226
         10136       181
         10140      1702
         10141        66
         10146        81
         10154        28
         10155       135
         10157       110
         10158       229
         10165         9
         10170        26
         10185       265
         10208       213
         10257       688
         10268        54
         10279       287
         10299      1148
         10333        48
         10372        55
         10397     30750
         10408       248
         10423      3764
         10431       263
         10434       106
         10469       113
         10529      1613
         10551        79
         10561       172
         10577        58
         10581        25
                   ...
         SRQ        3318
         STC          77
         STL       46181
         STT        4171
         STX         932
         SUN         867
         SUX         589
         SWF         678
         SYR        5447
         TLH        3141
         TOL         897
         TPA       63077
         TRI        1906
         TTN        2771
         TUL       13701
         TUS       14922
         TVC        2660
         TWF         805
         TXK         918
         TYR        2199
         TYS        6754
         UST         144
         VEL         200
         VLD         925
         VPS        4744
         WRG         649
         WYS         208
         XNA        8963
         YAK         650
         YUM        1854
         Length: 929, dtype: int64
```

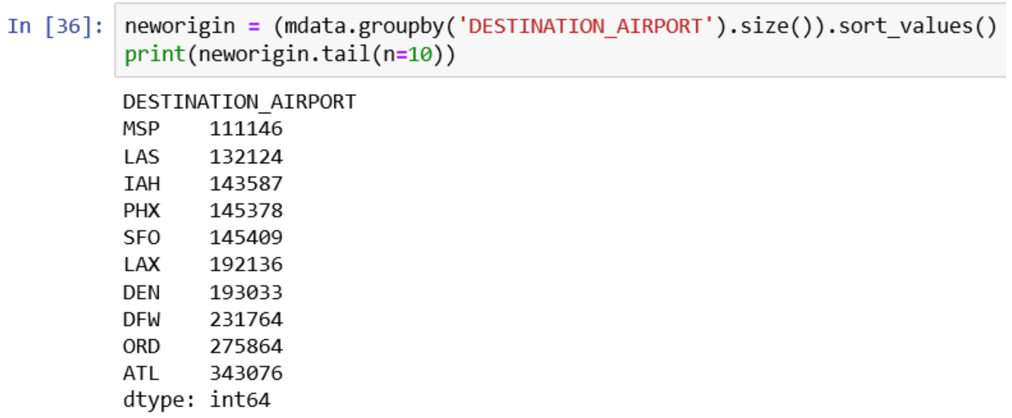


```
In [34]: neworigin = (mdata.groupby('ORIGIN_AIRPORT').size()).sort_values()
         print(neworigin.tail(n=10))

         ORIGIN_AIRPORT
         MSP    111055
         LAS    131937
         IAH    144019
         SFO    145491
         PHX    145552
         LAX    192003
         DEN    193402
         DFW    232647
         ORD    276554
         ATL    343506
         dtype: int64
```

```
In [35]: (mdata.groupby('DESTINATION_AIRPORT').size())

Out[35]: DESTINATION_AIRPORT
         10135     224
         10136     183
         10140    1706
         10141      67
         10146      82
         10154      27
         10155     135
         10157     110
         10158     230
         10165       9
         10170      26
         10185     266
         10208     212
         10257     692
         10268      54
         10279     287
         10299    1148
         10333      47
         10372      54
```



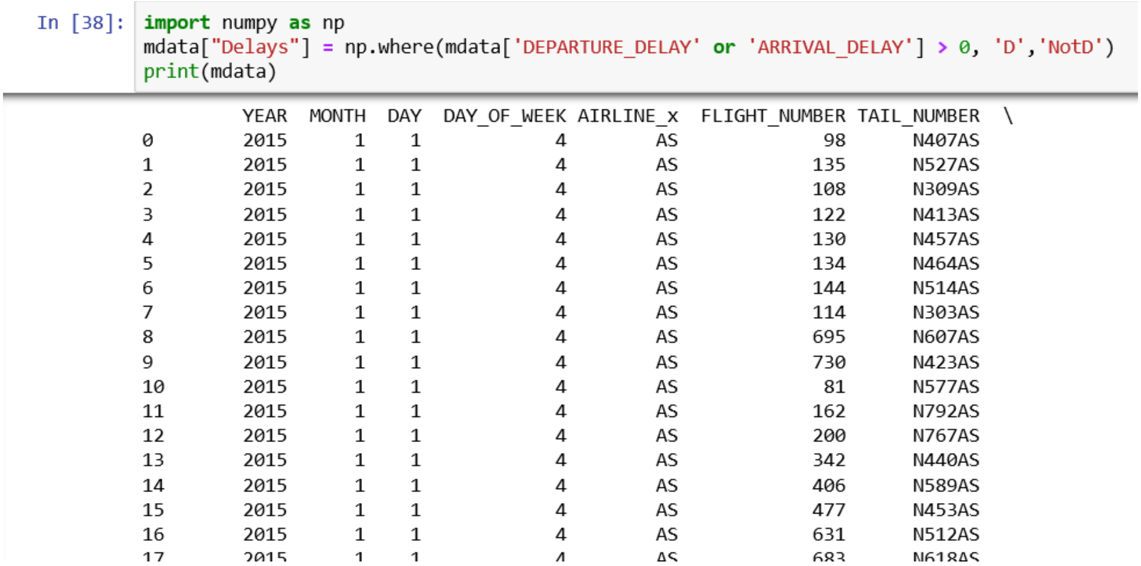
```
STL       46273
STT        4306
STX         933
SUN         854
SUX         591
SWF         680
SYR        5475
TLH        3149
TOL         901
TPA       63157
TRI        1912
TTN        2761
TUL       13748
TUS       14956
TVC        2666
TWF         806
TXK         915
TYR        2199
TYS        6764
UST         146
VEL         197
VLD         921
VPS        4743
WRG         652
WYS         207
XNA        8986
YAK         652
YUM        1856
Length: 929, dtype: int64
```



```
In [36]: neworigin = (mdata.groupby('DESTINATION_AIRPORT').size()).sort_values()
         print(neworigin.tail(n=10))
```

```
DESTINATION_AIRPORT
MSP    111146
LAS    132124
IAH    143587
PHX    145378
SFO    145409
LAX    192136
DEN    193033
DFW    231764
ORD    275864
ATL    343076
dtype: int64
```

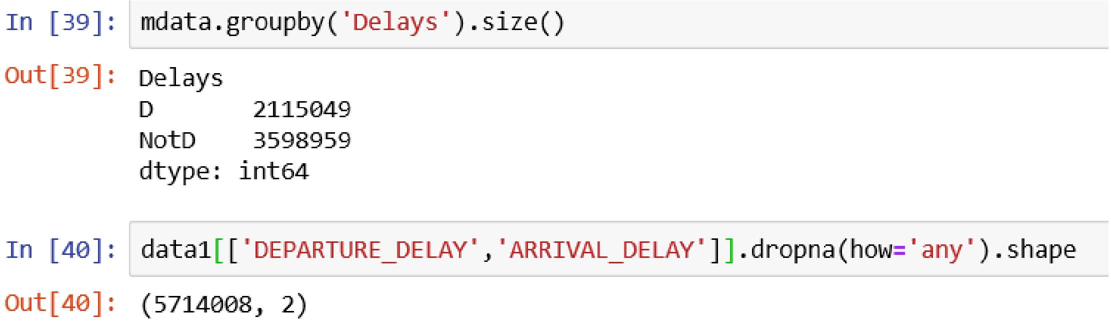
13. Create the new column; if the flight has delay on departure or arrival then the value will be ‘Delay’ . If not, ‘Not Delay’.



```
In [38]: import numpy as np
         mdata["Delays"] = np.where(mdata['DEPARTURE_DELAY' or 'ARRIVAL_DELAY'] > 0, 'D','NotD')
         print(mdata)
```

```
                YEAR  MONTH  DAY  DAY_OF_WEEK AIRLINE_x  FLIGHT_NUMBER TAIL_NUMBER  \
0               2015      1    1            4        AS             98      N407AS
1               2015      1    1            4        AS            135      N527AS
2               2015      1    1            4        AS            108      N309AS
3               2015      1    1            4        AS            122      N413AS
4               2015      1    1            4        AS            130      N457AS
5               2015      1    1            4        AS            134      N464AS
6               2015      1    1            4        AS            144      N514AS
7               2015      1    1            4        AS            114      N303AS
8               2015      1    1            4        AS            695      N607AS
9               2015      1    1            4        AS            730      N423AS
10              2015      1    1            4        AS             81      N577AS
11              2015      1    1            4        AS            162      N792AS
12              2015      1    1            4        AS            200      N767AS
13              2015      1    1            4        AS            342      N440AS
14              2015      1    1            4        AS            406      N589AS
15              2015      1    1            4        AS            477      N453AS
16              2015      1    1            4        AS            631      N512AS
17              2015      1    1            4        AS            683      N618AS
```

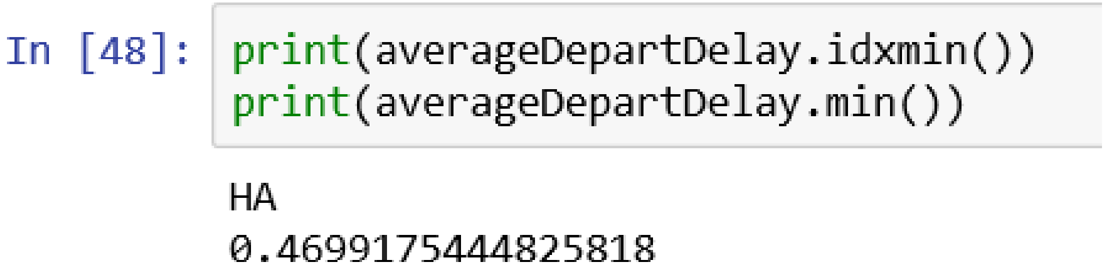
14. How many flights that are delay, how many flight that are not delay?



```
In [39]:  mdata.groupby('Delays').size()

Out[39]:  Delays
          D        2115049
          NotD     3598959
          dtype: int64

In [40]:  data1[['DEPARTURE_DELAY','ARRIVAL_DELAY']].dropna(how='any').shape

Out[40]:  (5714008, 2)
```

15. Create one of your own insight from the data

If Chu want to avoid delay flight. Which airline Chu should to choice? And what is average delay time?



```
In [48]: print(averageDepartDelay.idxmin())
         print(averageDepartDelay.min())

         HA
         0.4699175444825818
```