Big Data Analytics Techniques and Applications

Homework 2

309552063 吳冠潔

Q1. Find the maximal delays (you should consider both ArrDelay and DepDelay) for each month of 2007.

Result:

Month	**1**	**2**	**3**	**4**	**5**	**6**
MaxDelay	1460	1359	1564	1415	1429	1360

Month	**7**	**8**	**9**	**10**	**11**	**12**
MaxDelay	1386	1472	1589	2601	1146	1956

Method:

一開始先將 2007 年的資料拿出來, 並存成 spark 的 dataframe, 並用 printSchema() 確認其結構。

```
file = path + '/2007.csv'
df = spark.read.csv(file, header=True)
# print(df.count())
df.printSchema()
```

```
root
 |-- Year: string (nullable = true)
 |-- Month: string (nullable = true)
 |-- DayofMonth: string (nullable = true)
 |-- DayOfWeek: string (nullable = true)
 |-- DepTime: string (nullable = true)
 |-- CRSDepTime: string (nullable = true)
 |-- ArrTime: string (nullable = true)
 |-- CRSArrTime: string (nullable = true)
 |-- UniqueCarrier: string (nullable = true)
 |-- FlightNum: string (nullable = true)
 |-- TailNum: string (nullable = true)
 |-- ActualElapsedTime: string (nullable = true)
 |-- CRSElapsedTime: string (nullable = true)
 |-- AirTime: string (nullable = true)
 |-- ArrDelay: string (nullable = true)
 |-- DepDelay: string (nullable = true)
 |-- Origin: string (nullable = true)
 |-- Dest: string (nullable = true)
 |-- Distance: string (nullable = true)
 |-- TaxiIn: string (nullable = true)
 |-- TaxiOut: string (nullable = true)
 |-- Cancelled: string (nullable = true)
 |-- CancellationCode: string (nullable = true)
 |-- Diverted: string (nullable = true)
 |-- CarrierDelay: string (nullable = true)
 |-- WeatherDelay: string (nullable = true)
 |-- NASDelay: string (nullable = true)
 |-- SecurityDelay: string (nullable = true)
 |-- LateAircraftDelay: string (nullable = true)
```

因為需要看 ArrDelay 及 DepDelay，但從上面的 Schema 還看現在的 ArrDelay 和 DepDelay 的型態為 string，所以先把他們轉成 integer，用 IntegerType()將其轉成 integer，並印出結構確認其型態。

```
df = df.withColumn('ArrDelay', df['ArrDelay'].cast(IntegerType()))
df = df.withColumn('DepDelay', df['DepDelay'].cast(IntegerType()))

df.printSchema()
```

```
root
 |-- Year: string (nullable = true)
 |-- Month: string (nullable = true)
 |-- DayofMonth: string (nullable = true)
 |-- DayOfWeek: string (nullable = true)
 |-- DepTime: string (nullable = true)
 |-- CRSDepTime: string (nullable = true)
 |-- ArrTime: string (nullable = true)
 |-- CRSArrTime: string (nullable = true)
 |-- UniqueCarrier: string (nullable = true)
 |-- FlightNum: string (nullable = true)
 |-- TailNum: string (nullable = true)
 |-- ActualElapsedTime: string (nullable = true)
 |-- CRSElapsedTime: string (nullable = true)
 |-- AirTime: string (nullable = true)
 |-- ArrDelay: integer (nullable = true)
 |-- DepDelay: integer (nullable = true)
 |-- Origin: string (nullable = true)
 |-- Dest: string (nullable = true)
 |-- Distance: string (nullable = true)
 |-- TaxiIn: string (nullable = true)
 |-- TaxiOut: string (nullable = true)
 |-- Cancelled: string (nullable = true)
 |-- CancellationCode: string (nullable = true)
 |-- Diverted: string (nullable = true)
 |-- CarrierDelay: string (nullable = true)
 |-- WeatherDelay: string (nullable = true)
 |-- NASDelay: string (nullable = true)
 |-- SecurityDelay: string (nullable = true)
 |-- LateAircraftDelay: string (nullable = true)
```

從結構中可以看到 ArrDelay 和 DepDelay 都轉成 integer。

再來先用 groupby.agg 取出每個月最大的的 ArrDelay 和 DepDelay，並存成新的 dataframe，命名為 dfDelay。在 dfDelay 中，再新增一個 column，其中存的是比較 MaxArrDelay 和 MaxDepDelay 後取出的最大值。

```
dfDelay =
df.groupby('Month').agg(F.max('ArrDelay').alias('MaxArrDelay'), \
                        F.max('DepDelay').alias('MaxDepDelay'))
minf = lit(float("-inf"))
delaymax = greatest(*[coalesce(col(x), minf) for x in
['MaxArrDelay', 'MaxDepDelay']])
dfDelay.withColumn("MaxDelay", delaymax).show()
```

```
+-----+----------+----------+--------+
|Month|MaxArrDelay|MaxDepDelay|MaxDelay|
+-----+----------+----------+--------+
|    3|      1564|      1547|  1564.0|
|    1|      1426|      1406|  1426.0|
|    2|      1359|      1340|  1359.0|
|    5|      1429|      1416|  1429.0|
|    4|      1402|      1415|  1415.0|
|    7|      1386|      1369|  1386.0|
|    6|      1351|      1360|  1360.0|
|    8|      1472|      1449|  1472.0|
|    9|      1665|      1689|  1689.0|
|   10|      2598|      2601|  2601.0|
|   11|      1146|      1137|  1146.0|
|   12|      1942|      1956|  1956.0|
+-----+----------+----------+--------+
```

從印出的 dataframe 中可以看到 MaxDelay 即為所求的答案。

Other:

在取 ArrDelay 和 DepDelay 時，一開始我沒有用 groupby，而是用 filter 去取出每個月的資料，再用 agg 得到個別的最大值，雖然有成功得到答案，但是速度相比使用 groupby 慢很多。

```
for m in range(1, 13):
    new_df = df.filter(df['Month'] == str(m))
    max_arrdelay = new_df.agg({"ArrDelay": "max"}).collect()[0][0]
    max_depdelay = new_df.agg({"DepDelay": "max"}).collect()[0][0]

    maxDelay = max(max_arrdelay, max_depdelay)

    print('2017/{} MaxDelay:{}, with ArrDelay:{} and
DepDelay:{}'.format(m, \
          maxDelay, max_arrdelay, max_depdelay))
```

```
2017/1 MaxDelay:1426, with ArrDelay:1426 and DepDelay:1406
2017/2 MaxDelay:1359, with ArrDelay:1359 and DepDelay:1340
2017/3 MaxDelay:1564, with ArrDelay:1564 and DepDelay:1547
2017/4 MaxDelay:1415, with ArrDelay:1402 and DepDelay:1415
2017/5 MaxDelay:1429, with ArrDelay:1429 and DepDelay:1416
2017/6 MaxDelay:1360, with ArrDelay:1351 and DepDelay:1360
2017/7 MaxDelay:1386, with ArrDelay:1386 and DepDelay:1369
2017/8 MaxDelay:1472, with ArrDelay:1472 and DepDelay:1449
2017/9 MaxDelay:1689, with ArrDelay:1665 and DepDelay:1689
2017/10 MaxDelay:2601, with ArrDelay:2598 and DepDelay:2601
2017/11 MaxDelay:1146, with ArrDelay:1146 and DepDelay:1137
2017/12 MaxDelay:1956, with ArrDelay:1942 and DepDelay:1956
```

Q2. How many flights were delayed caused by security between 2000 ~ 2005? Please show the counting for each year.

Result:

Year	Count of delayed caused by security
2000	0
2001	0
2002	0
2003	3740
2004	8158
2005	6627

Method:

一開始先將 2000-2015 的資料放到 dataframe 中。

```
files = []
for i in range(2000, 2005+1):
    files.append(path + str(i) + '.csv')
print(files)
df = spark.read.csv(files, header=True)
```

因為要看的是有無發生 Security Delay，我用的是看 SecurityDelay 該欄有無大於零來做判斷，先將其轉成 Integer 型態，接著設好條件，當 SecurityDelay > 0 就表示有因為 Security 造成的 Delay，設為 1，反之則設為 0，最後使用該條件在 dataframe 中新增一欄 ***HasSecDelay***。

```
df = df.withColumn('SecurityDelay',
df['SecurityDelay'].cast(IntegerType()))
new_column = when(col('SecurityDelay') > 0, 1).otherwise(0)

df = df.withColumn('HasSecDelay', new_column)
df.show()
```

```
+----+-----+----------+---------+-------+-----------+-------+-----------+-------------+---------+-----------+----------------+----------------+-------+--------+--------+
|Year|Month|DayofMonth|DayOfWeek|DepTime|CRSDepTime|ArrTime|CRSArrTime|UniqueCarrier|FlightNum|TailNum|ActualElapsedTime|CRSElapsedTime|AirTime|ArrDelay|DepDelay|
+----+-----+----------+---------+-------+-----------+-------+-----------+-------------+---------+-----------+----------------+----------------+-------+--------+--------+
|2000|    1|        28|        5|   1647|       1647|   1906|       1859|           HP|      154|  N808AW|             259|             252|    233|       7|       0|
|2000|    1|        29|        6|   1648|       1647|   1939|       1859|           HP|      154|  N653AW|             291|             252|    239|      40|       1|
|2000|    1|        30|        7|     NA|       1647|     NA|       1859|           HP|      154|  N801AW|              NA|             252|     NA|      NA|      NA|
|2000|    1|        31|        1|   1645|       1647|   1852|       1859|           HP|      154|  N806AW|             247|             252|    226|      -7|      -2|
|2000|    1|         1|        6|    842|        846|   1057|       1101|           HP|      609|  N158AW|             255|             255|    244|      -4|      -4|
|2000|    1|         2|        7|    849|        846|   1148|       1101|           HP|      609|  N656AW|             299|             255|    267|      47|       3|
|2000|    1|         3|        1|    844|        846|   1121|       1101|           HP|      609|  N803AW|             277|             255|    244|      20|      -2|
|2000|    1|         1|        6|   1702|       1657|   1912|       1908|           HP|      611|  N652AW|             250|             251|    232|       4|       5|
|2000|    1|         2|        7|   1658|       1657|   1901|       1908|           HP|      611|  N807AW|             243|             251|    233|      -7|       1|
|2000|    1|         3|        1|   1656|       1657|   1922|       1908|           HP|      611|  N807AW|             266|             251|    241|      14|      -1|
|2000|    1|         4|        2|   1955|       1932|   2230|       2153|           HP|      613|  N509DC|             275|             261|    232|      37|      23|
|2000|    1|         5|        3|   1934|       1932|   2133|       2153|           HP|      613|  N509DC|             239|             261|    224|     -20|       2|
|2000|    1|         6|        4|   1929|       1932|   2125|       2153|           HP|      613|  N303AW|             236|             261|    220|     -28|      -3|
|2000|    1|         7|        5|   1932|       1932|   2146|       2153|           HP|      613|  N173AW|             254|             261|    237|      -7|       0|
|2000|    1|         9|        7|   2008|       1932|   2221|       2153|           HP|      613|  N168AW|             253|             261|    237|      28|      36|
|2000|    1|        10|        1|   1926|       1932|   2147|       2153|           HP|      613|  N160AW|             261|             261|    235|      -6|      -6|
|2000|    1|        11|        2|   1932|       1932|   2126|       2153|           HP|      613|  N160AW|             234|             261|    217|     -27|       0|
|2000|    1|        12|        3|   1936|       1932|   2142|       2153|           HP|      613|  N322AW|             246|             261|    227|     -11|       4|
|2000|    1|        13|        4|   1942|       1932|   2153|       2153|           HP|      613|  N160AW|             251|             261|    220|       0|      10|
|2000|    1|        14|        5|   1932|       1932|   2131|       2153|           HP|      613|  N314AW|             239|             261|    218|     -22|       0|
+----+-----+----------+---------+-------+-----------+-------+-----------+-------------+---------+-----------+----------------+----------------+-------+--------+--------+
only showing top 20 rows
```

DepDelay	Origin	Dest	Distance	TaxiIn	TaxiOut	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay	HasSecDelay
0	ATL	PHX	1587	15	11	0	NA	0	NA	NA	NA	null	NA	0
1	ATL	PHX	1587	5	47	0	NA	0	NA	NA	NA	null	NA	0
NA	ATL	PHX	1587	0	0	1	NA	0	NA	NA	NA	null	NA	0
-2	ATL	PHX	1587	7	14	0	NA	0	NA	NA	NA	null	NA	0
-4	ATL	PHX	1587	3	8	0	NA	0	NA	NA	NA	null	NA	0
3	ATL	PHX	1587	8	24	0	NA	0	NA	NA	NA	null	NA	0
-2	ATL	PHX	1587	6	27	0	NA	0	NA	NA	NA	null	NA	0
5	ATL	PHX	1587	5	13	0	NA	0	NA	NA	NA	null	NA	0
1	ATL	PHX	1587	3	7	0	NA	0	NA	NA	NA	null	NA	0
-1	ATL	PHX	1587	5	20	0	NA	0	NA	NA	NA	null	NA	0
23	ATL	PHX	1587	5	38	0	NA	0	NA	NA	NA	null	NA	0
2	ATL	PHX	1587	5	10	0	NA	0	NA	NA	NA	null	NA	0
-3	ATL	PHX	1587	5	11	0	NA	0	NA	NA	NA	null	NA	0
0	ATL	PHX	1587	4	13	0	NA	0	NA	NA	NA	null	NA	0
36	ATL	PHX	1587	4	12	0	NA	0	NA	NA	NA	null	NA	0
-6	ATL	PHX	1587	7	19	0	NA	0	NA	NA	NA	null	NA	0
0	ATL	PHX	1587	6	11	0	NA	0	NA	NA	NA	null	NA	0
4	ATL	PHX	1587	7	12	0	NA	0	NA	NA	NA	null	NA	0
10	ATL	PHX	1587	5	26	0	NA	0	NA	NA	NA	null	NA	0
0	ATL	PHX	1587	6	15	0	NA	0	NA	NA	NA	null	NA	0

從印出的 dataframe 中可以看到在最後面新增了一欄 HasSecDelay。

最後根據新的欄位 HasSecDelay，用 groupby 區分不同的年份，再將各年份的 HasSecDelay 加總有幾次 Delay 是因為 Security 的問題造成的 Delay。

```
df.groupby('Year').agg({'HasSecDelay': 'sum'}).show()
```

Year	sum(HasSecDelay)
2000	0
2001	0
2002	0
2003	3740
2004	8158
2005	6627

從上圖中可以看到，其計算不同年時發生 delay 的次數。

Other:

在判斷有無發生 Security Delay 的部分，我除了看 SecurityDelay 有無大於 0，因為想到在第一題中，在印出 dataframe 的前幾筆資料時，在 ArrDelay 和 DepDelay 中，有時會有 Delay 負數的情況，所以我有試過用不同的條件，將只要 SecurityDelay 那欄不為 0 且不為空的時候，就當作有發生 Security Delay，在嘗試這樣的條件式後得到的結果和只看 SecurityDelay 大於 0 的情況一樣，代表 SecurityDelay 中沒有為負數的資料，所以最後我還是選擇用 SecurityDelay>0 的條件，去得出結果。

Q3. List Top 5 airports which occur delays most and least in 2008. (Please show the IATA airport code)

Result:

	airport that occur delays most	airport that occur delays least
1	ATL	PUB
2	ORD	TUP
3	DFW	PIR
4	DEN	INL
5	LAX	BJI

Method:

一開始先將 2008 的資料存成 dataframe。

```
file = path + '2008.csv'
df = spark.read.csv(file, header=True)
df.printSchema()
```

```
root
 |-- Year: string (nullable = true)
 |-- Month: string (nullable = true)
 |-- DayofMonth: string (nullable = true)
 |-- DayOfWeek: string (nullable = true)
 |-- DepTime: string (nullable = true)
 |-- CRSDepTime: string (nullable = true)
 |-- ArrTime: string (nullable = true)
 |-- CRSArrTime: string (nullable = true)
 |-- UniqueCarrier: string (nullable = true)
 |-- FlightNum: string (nullable = true)
 |-- TailNum: string (nullable = true)
 |-- ActualElapsedTime: string (nullable = true)
 |-- CRSElapsedTime: string (nullable = true)
 |-- AirTime: string (nullable = true)
 |-- ArrDelay: string (nullable = true)
 |-- DepDelay: string (nullable = true)
 |-- Origin: string (nullable = true)
 |-- Dest: string (nullable = true)
 |-- Distance: string (nullable = true)
 |-- TaxiIn: string (nullable = true)
 |-- TaxiOut: string (nullable = true)
 |-- Cancelled: string (nullable = true)
 |-- CancellationCode: string (nullable = true)
 |-- Diverted: string (nullable = true)
 |-- CarrierDelay: string (nullable = true)
 |-- WeatherDelay: string (nullable = true)
 |-- NASDelay: string (nullable = true)
 |-- SecurityDelay: string (nullable = true)
 |-- LateAircraftDelay: string (nullable = true)
```

接著看 ArrDelay 和 DepDelay，先將其從 String 轉成 integer，當 ArrDelay 或 DepDelay 其中一個大於零時，就當作其有發生 Delay，因為有時會有小於零的資料，但我認為提早出發和提早抵達不算是發生 delay，所以只取大於零時，設定好條件後，在原本的 dataframe 中新增一個 column，命名為 ***HasDelay***，當符合條件時為 1，否則為 0。

```
df = df.withColumn('ArrDelay', df['ArrDelay'].cast(IntegerType()))
df = df.withColumn('DepDelay', df['DepDelay'].cast(IntegerType()))
```

```
new_column = when((col('ArrDelay') > 0) | \
                  (col('DepDelay') > 0), 1).otherwise(0)

df1 = df.withColumn('HasDelay', new_column)
df1.show()
```

Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNum	TailNum	ActualElapsedTime	CRSElapsedTime	AirTime	ArrDelay	DepDelay	C
2008	1	3	4	2003	1955	2211	2225	WN	335	N712SW	128	150	116	-14	8	
2008	1	3	4	754	735	1002	1000	WN	3231	N772SW	128	145	113	2	19	
2008	1	3	4	628	620	804	750	WN	448	N428WN	96	90	76	14	8	
2008	1	3	4	926	930	1054	1100	WN	1746	N612SW	88	90	78	-6	-4	
2008	1	3	4	1829	1755	1959	1925	WN	3920	N464WN	90	90	77	34	34	
2008	1	3	4	1940	1915	2121	2110	WN	378	N726SW	101	115	87	11	25	
2008	1	3	4	1937	1830	2037	1940	WN	509	N763SW	240	250	230	57	67	
2008	1	3	4	1039	1040	1132	1150	WN	535	N428WN	233	250	219	-18	-1	
2008	1	3	4	617	615	652	650	WN	11	N689SW	95	95	70	2	2	
2008	1	3	4	1620	1620	1639	1655	WN	810	N648SW	79	95	70	-16	0	
2008	1	3	4	706	700	916	915	WN	100	N690SW	130	135	106	1	6	
2008	1	3	4	1644	1510	1845	1725	WN	1333	N334SW	121	135	107	80	94	
2008	1	3	4	1426	1430	1426	1425	WN	829	N476WN	60	55	39	1	-4	
2008	1	3	4	715	715	720	710	WN	1016	N765SW	65	55	37	10	0	
2008	1	3	4	1702	1700	1651	1655	WN	1827	N420WN	49	55	35	-4	2	
2008	1	3	4	1029	1020	1021	1010	WN	2272	N263WN	52	50	37	11	9	
2008	1	3	4	1452	1425	1640	1625	WN	675	N286WN	228	240	213	15	27	
2008	1	3	4	754	745	940	955	WN	1144	N778SW	226	250	205	-15	9	
2008	1	3	4	1323	1255	1526	1510	WN	4	N674AA	123	135	110	16	28	
2008	1	3	4	1416	1325	1512	1435	WN	54	N643SW	56	70	49	37	51	

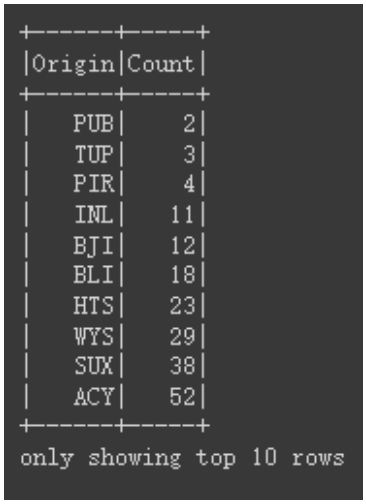
only showing top 20 rows

DepDelay	Origin	Dest	Distance	TaxiIn	TaxiOut	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay	HasDelay
8	IAD	TPA	810	4	8	0	null	0	NA	NA	NA	NA	NA	1
19	IAD	TPA	810	5	10	0	null	0	NA	NA	NA	NA	NA	1
8	IND	BWI	515	3	17	0	null	0	NA	NA	NA	NA	NA	1
-4	IND	BWI	515	3	7	0	null	0	NA	NA	NA	NA	NA	0
34	IND	BWI	515	3	10	0	null	0	2	0	0	0	32	1
25	IND	JAX	688	4	10	0	null	0	NA	NA	NA	NA	NA	1
67	IND	LAS	1591	3	7	0	null	0	10	0	0	0	47	1
-1	IND	LAS	1591	7	7	0	null	0	NA	NA	NA	NA	NA	0
2	IND	MCI	451	6	19	0	null	0	NA	NA	NA	NA	NA	1
0	IND	MCI	451	3	6	0	null	0	NA	NA	NA	NA	NA	0
6	IND	MCO	828	5	19	0	null	0	NA	NA	NA	NA	NA	1
94	IND	MCO	828	6	8	0	null	0	8	0	0	0	72	1
-4	IND	MDW	162	9	12	0	null	0	NA	NA	NA	NA	NA	1
0	IND	MDW	162	7	21	0	null	0	NA	NA	NA	NA	NA	1
2	IND	MDW	162	4	10	0	null	0	NA	NA	NA	NA	NA	1
9	IND	MDW	162	6	9	0	null	0	NA	NA	NA	NA	NA	1
27	IND	PHX	1489	7	8	0	null	0	3	0	0	0	12	1
9	IND	PHX	1489	5	16	0	null	0	NA	NA	NA	NA	NA	1
28	IND	TPA	838	4	9	0	null	0	0	0	0	0	16	1
51	ISP	BWI	220	2	5	0	null	0	12	0	0	0	25	1

最後用 groupby.agg 得到以 Origin airport 做計算的加總 delay 次數，但因為得出的結果是亂序，所以再用 orderBy 排序，因為需要最多的五個以及最小的五個，所以我讓其排序兩次，一次大到小，另一次從小到大排序，我印出個別的前 10 個來看，因為問題只問 top 5，在最後的答案我只取個別前五名。

Origin	Count
ATL	237091
ORD	197937
DFW	160519
DEN	134941
LAX	112426
IAH	109517
PHX	105595
LAS	92915
EWR	89004
DTW	88197

only showing top 10 rows



```
+------+-----+
|Origin|Count|
+------+-----+
|   PUB|    2|
|   TUP|    3|
|   PIR|    4|
|   INL|   11|
|   BJI|   12|
|   BLI|   18|
|   HTS|   23|
|   WYS|   29|
|   SUX|   38|
|   ACY|   52|
+------+-----+
only showing top 10 rows
```

上面兩張圖中可以看到其分別為大到小與小到大排序後的前 10 個結果，其中的個別前五個就為這題的結果。

Other:

在看怎樣才能算是有 Delay 時，考慮了兩種可能，第一種為只要在 ArrDelay/DepDelay 中不為零，就算是發生 Delay；第二種為當 CarrierDelay, WeatherDelay, NASDelay, NASDelay 及 LateAircraftDelay 其中之一有數值才算是發生 Delay。但是因為在 ArrDelay/DepDelay 有數值時, CarrierDelay, WeatherDelay, NASDelay, NASDelay 及 LateAircraftDelay 不一定會有紀錄，所以後來決定用第一種當作看有無發生 Delay 的依據。

另外我本來將 ArrDelay 和 DepDelay 分開看，並將其分別當作是目的地和出發機場個別的 delay，但後來想一想覺得應該都要算是出發機場的 delay。