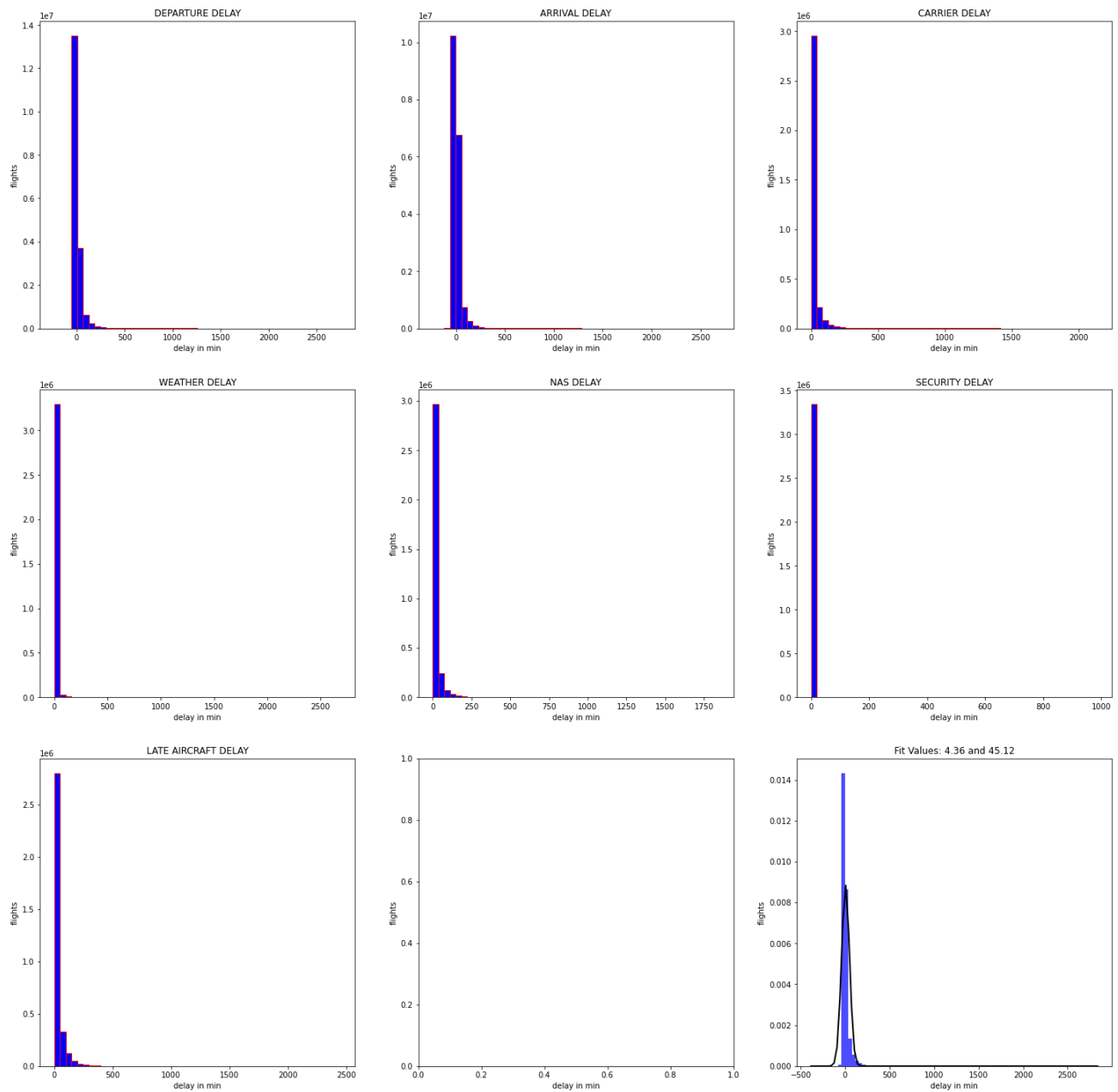


Python script (rb.py) for generating answers to the questions 1-5 is located at <https://gitfront.io/r/user-1008619/zQZGJNbqebWC/DelayTimeAnalys/>

Output from the script:

```
In [35]: runfile('D:/RB/rb.py', wdir='D:/RB')
Mladjan Jovanovic - Delay Time Analys ver 1.0
~~Mean delay time in minutes:
DEP_DELAY 9.580880
ARR_DELAY 4.361594
CARRIER_DELAY 19.832233
WEATHER_DELAY 3.085360
NAS_DELAY 15.586749
SECURITY_DELAY 0.090433
LATE_AIRCRAFT_DELAY 25.117133
dtype: float64
~~Median delay time in minutes:
DEP_DELAY -2.0
ARR_DELAY -6.0
CARRIER_DELAY 1.0
WEATHER_DELAY 0.0
NAS_DELAY 2.0
SECURITY_DELAY 0.0
LATE_AIRCRAFT_DELAY 3.0
dtype: float64
~~~~~
Total mean delay time in minutes: 11.09
Total median delay time in minutes: -0.29
```

Histograms of Delay's



Looking at the mean and median value we can conclude that the delay time does not have a symmetric bell shape.

Because mean is higher than the median we can conclude that we have major outliers with big delay time in minutes that skew delay time in +.

This is also proven by looking at the skew value

Overall, in this year positive and negative delay times off all flights are annul i.e. ~ 0 min.

~~~~~

```

~~Skew delay time:
DEP_DELAY 9.555269
ARR_DELAY 8.368851
CARRIER_DELAY 9.456533
WEATHER_DELAY 21.228857
NAS_DELAY 9.200847
SECURITY_DELAY 103.882829
LATE_AIRCRAFT_DELAY 5.063882
dtype: float64
~~Kurt delay time:
DEP_DELAY 173.174813
ARR_DELAY 143.049336
CARRIER_DELAY 135.947272
WEATHER_DELAY 689.761446
NAS_DELAY 191.878263
SECURITY_DELAY 20862.318002
LATE_AIRCRAFT_DELAY 59.302456
dtype: float64
~~~~~
Total skew delay time: 23.82
Total kurt delay time: 3179.35

```

Looking at the skew value (horizontal push) we can conclude that the distribution is moved to left  
and we have big kurt value (vertical push) so we have very big peak on that graph.  
~~~~~

Looking at the histograms we can conclude that we have similar curve for all type of delays

We can represent Arrival Delay as normal distribution

D:\RB\rb.py:109: FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated. Please use Series.dt.isocalendar().week instead.

```
DWYR=df.groupby(df['FL_DATE'].dt.weekofyear)['ARR_DELAY'].agg(['mean', 'median'])
mean median
```

```

FL_DATE
2016 3.519290 -6.0
2017 4.326357 -6.0
2018 5.048581 -6.0 mean median
FL_DATE
2016Q1 1.837386 -7.0
2016Q2 3.416893 -6.0
2016Q3 6.453750 -5.0
2016Q4 2.176429 -6.0
2017Q1 4.083264 -6.0
2017Q2 7.504681 -5.0
2017Q3 5.026123 -6.0
2017Q4 0.552357 -7.0
2018Q1 2.807642 -7.0
2018Q2 6.014243 -5.0
2018Q3 7.330627 -5.0
2018Q4 3.790801 -6.0 mean median
FL_DATE
1 3.709700 -6.0
2 1.830092 -7.0
3 3.077865 -6.0

```

4 3.858950 -6.0  
5 4.807768 -5.0  
6 8.284220 -4.0  
7 9.239021 -4.0  
8 8.274225 -5.0  
9 1.072023 -7.0  
10 1.403745 -7.0  
11 0.521142 -7.0  
12 5.027310 -5.0 mean median

FL\_DATE

1 10.550293 -3.0  
2 5.048188 -6.0  
3 4.022601 -6.0  
4 -2.824308 -9.0  
5 -1.441987 -9.0  
6 4.404911 -6.0  
7 1.411684 -8.0  
8 2.707095 -7.0  
9 1.619296 -7.0  
10 3.079752 -6.0  
11 2.804365 -5.0  
12 2.928082 -6.0  
13 3.914476 -6.0  
14 9.903865 -4.0  
15 0.786126 -7.0  
16 1.615487 -7.0  
17 3.399284 -6.0  
18 3.469812 -6.0  
19 1.520414 -7.0  
20 7.395439 -4.0  
21 5.821319 -5.0  
22 6.412273 -6.0  
23 5.821949 -5.0  
24 8.174487 -4.0  
25 10.852063 -3.0  
26 8.444586 -4.0  
27 4.179288 -7.0  
28 8.511256 -5.0  
29 12.653694 -3.0  
30 12.236237 -3.0  
31 11.132332 -3.0  
32 12.802204 -3.0  
33 10.477433 -4.0  
34 2.174972 -7.0  
35 1.243282 -7.0  
36 1.731704 -7.0  
37 -0.461546 -8.0  
38 2.224125 -7.0  
39 0.824624 -7.0  
40 -0.049848 -7.0  
41 4.696889 -6.0  
42 0.834736 -6.0  
43 1.892175 -6.0  
44 -1.360817 -8.0  
45 -0.634956 -7.0

```

46 3.232003 -6.0
47 -1.755826 -8.0
48 1.083976 -8.0
49 1.875849 -6.0
50 7.188354 -5.0
51 5.551701 -5.0
52 8.016084 -4.0
53 2.431287 -6.0 mean median
FL_DATE
0 5.444530 -6.0
1 3.316338 -6.0
2 3.142433 -6.0
3 5.927686 -5.0
4 6.642406 -5.0
5 1.710869 -7.0
6 3.773729 -6.0 mean median
OP_CARRIER
9E 4.452778 -9.0
AA 4.795570 -5.0
AS -0.653152 -6.0
B6 10.194332 -5.0
DL -0.307096 -9.0
EV 6.946690 -7.0
F9 9.512646 -5.0
G4 9.983518 -4.0
HA 0.907382 -3.0
MQ 5.357280 -5.0
NK 6.527232 -6.0
OH 8.237351 -3.0
OO 6.247458 -6.0
UA 3.174260 -8.0
VX 8.437540 -2.0
WN 4.493939 -4.0
YV 8.853066 -4.0
YX 3.076687 -8.0 mean median
ORIGIN
ABE 6.647177 -6.0
ABI 7.055581 -4.0
ABQ 3.236020 -5.0
ABR 4.829603 -6.0
ABY 9.399847 -6.0
... ..
WYS 5.211288 -1.0
XNA 6.027394 -8.0
YAK -8.223541 -13.0
YNG 75.000000 75.0
YUM -2.317979 -8.0

[362 rows x 2 columns] mean median
DEST
ABE 5.300259 -6.0
ABI 6.304368 -3.0
ABQ 4.532681 -4.0
ABR -0.331810 -10.0
ABY 8.779231 -6.0

```

```
....
WYS 4.696221 3.0
XNA 5.239033 -6.0
YAK -3.830740 -10.0
YNG 59.500000 59.5
YUM -0.319176 -7.0
```

```
[360 rows x 2 columns]
```

```
~~~~~
```

```
Finished.
```

```
In [36]:
```