CS4740 Spring 2021 Cloud Computing PA#4

Name: Victoria Wang

UVa User ID: vxw6ta

1. [The five airlines with the best on-time performance and the five airlines with the worst on-time performance for Jan 2021] Cut-and-paste your mapper file, your reducer file, and a screenshot of the "cat" after your program has successfully executed on hadoop (e.g., "cat delays/part-00000 | more"). If you were unable to complete this part, show your mapper and your reducer and describe the problem (you were unable to debug).

Solution

Top Five Best Airlines (Jan 2021)

```
1)"HA"   2.5499843309307426
2)"AS"   4.045389801538316
3)"QX"   4.180262407301768
4)"WN"   4.2302144024295645
5)"F9"   5.304215419137682
```

Top Five Worst Airlines (Jan 2021)

```
1)"B6"   12.907216494845361
2)"G4"   12.029496923633731
3)"MQ"   10.104453441295547
4)"OO"   9.583526840595548
5)"YV"   9.50642883013401
```

OnTimePerfmapper.py

```
#!/usr/bin/env python3

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    line = line.split(",")
    # increase counters
    for airline in line:
        airline = line[1]
        delay = line[14]
        if delay == "":
            continue
        print ("{0}\t{1}".format(airline, delay))
```

OnTimePerfreducer.py

```
#!/usr/bin/env python3

from operator import itemgetter
import sys

airline_delay = {}

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    airline, delay = line.split('\t')
    # convert count (currently a string) to int

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if airline in airline_delay:
        airline_delay[airline].append(int(float(delay)))
    else:
        airline_delay[airline] = []
        airline_delay[airline].append(int(float(delay)))

for airline in airline_delay.keys():
    avg_delay = sum(airline_delay[airline])*1.0 / len(airline_delay[airline])
    print ("{0}\t{1}".format(airline, avg_delay))
```

cat delays/part-00000 | more

```
[ubuntu@ip-172-31-67-92:~$ cat delays/part-00000 | sort -k2 -n | more
"HA"    2.5499843309307426
"AS"    4.045389801538316
"QX"    4.180262407301768
"WN"    4.2302144024295645
"F9"    5.304215419137682
"UA"    6.034677215722582
"YX"    6.278044909201251
"9E"    6.334652752788773
"NK"    6.558481377942105
"DL"    6.782619542619543
"AA"    8.18221389753601
"OH"    9.439964011123834
"YV"    9.50642883013401
"OO"    9.583526840595548
"MQ"    10.104453441295547
"G4"    12.029496923633731
"B6"    12.907216494845361
```

2. [For each of the three worst airlines, the 15 worst routes for Jan 2021 (where the planes arrive the latest)] Cut-and-paste your mapper file, your reducer file, and a screenshot of the "cat" after your program has successfully executed on hadoop (e.g., "cat routes/part-00000 | more"). If you were unable to complete this part, show your mapper and your reducer and describe the problem (you were unable to debug).

Solution

"B6" Airline Worst Routes

```
1."B6"  "PHL"  "PBI"  191.0
2."B6"  "LAX"  "LAS"  93.0
3."B6"  "LAS"  "LAX"  87.5
4."B6"  "SEA"  "FLL"  68.0
5."B6"  "ORD"  "JFK"  64.66666666666667
6."B6"  "JAX"  "EWR"  62.81818181818182
7."B6"  "LAX"  "BZN"  60.0
8."B6"  "JFK"  "ORD"  53.666666666666664
9."B6"  "HPN"  "RSW"  52.352941176470 59
10. "B6"  "PBI"  "PHL"  50.75
11. "B6"  "RSW"  "HPN"  50.05714285714286
12. "B6"  "PBI"  "LGA"  50.0
13. "B6"  "PHX"  "JFK"  49.111111111111114
14. "B6"  "EWR"  "BOS"  48.375
15. "B6"  "FLL"  "AUS"  44.9
```

"G4" Airline Worst Routes

```
1."G4"  "GSO"  "SFB"  289.0
2."G4"  "LAX"  "BOI"  252.4
3."G4"  "LAS"  "EUG"  239.22222222222223
4."G4"  "GSP"  "FLL"  220.85714285714286
5."G4"  "FLL"  "GSP"  215.57142857142858
6."G4"  "IAG"  "PGD"  214.0
7."G4"  "BLI"  "OAK"  188.8
8."G4"  "SWF"  "SFB"  187.0
9."G4"  "MLI"  "PIE"  153.0
10. "G4"  "BLV"  "FLL"  151.0
11. "G4"  "FLL"  "BLV"  142.0
12. "G4"  "SFB"  "SWF"  139.0
13. "G4"  "ORF"  "FLL"  114.5
14. "G4"  "PGD"  "IAG"  101.5
15. "G4"  "HTS"  "PGD"  92.0
```

“MQ” Airline Worst Routes

```
1."MQ"  "PIA"  "ORD"  152.33333333333334
2."MQ"  "DFW"  "LAW"  62.58490566037736
3."MQ"  "ORD"  "PBI"  48.25
4."MQ"  "DCA"  "ORD"  47.0
5."MQ"  "ORD"  "ABE"  45.666666666666664
6."MQ"  "PBI"  "ORD"  45.25
7."MQ"  "PNS"  "MIA"  45.25
8."MQ"  "ORD"  "TOL"  44.0
9."MQ"  "DAY"  "ORD"  42.34615384615385
10.  "MQ"   "ICT"  "ORD"  41.70967741935484
11.  "MQ"   "LBB"  "DFW"  40.0
12.  "MQ"   "FAR"  "DFW"  38.47222222222222
13.  "MQ"   "MIA"  "PNS"  38.0
14.  "MQ"   "ABE"  "ORD"  37.666666666666664
15.  "MQ"   "BZN"  "ORD"  36.37931034482759
```

RouteMapper.py

```
#!/usr/bin/env python3

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    line = line.split(",")
    # increase counters
    for airline in line:
        airline = line[1]
        #print(airline,len(airline))
        origin = line[4]
        destination = line[9]
        delay = line[14]
        if delay == "":
            continue
        if airline == "\"B6\"" or airline == "\"G4\"" or airline == "\"MQ\"":
            print ("{0}\t{1}".format(airline + "," + origin + ","+
destination, delay))
```

RouteReducer.py

```
#!/usr/bin/env python3

from operator import itemgetter
import sys

airline_delay = {}

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    airline, delay = line.split('\t')
    # convert count (currently a string) to int

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if airline in airline_delay:
        airline_delay[airline].append(int(float(delay)))
    else:
        airline_delay[airline] = []
        airline_delay[airline].append(int(float(delay)))

for airline in airline_delay.keys():
    avg_delay = sum(airline_delay[airline])*1.0 / len(airline_delay[airline])
    airline, origin, dest = airline.split(',')
    print ("{0}\t{1}\t{2}\t{3}".format(airline, origin, dest, avg_delay))
```

cat routes/part-00000 | more

“B6” Airline

```
ubuntu@ip-172-31-67-92:~$ cat routes/part-00000 | sort -k1,1 -k4nr,4 | more
"B6"    "PHL"   "PBI"   191.0
"B6"    "LAX"   "LAS"   93.0
"B6"    "LAS"   "LAX"   87.5
"B6"    "SEA"   "FLL"   68.0
"B6"    "ORD"   "JFK"   64.66666666666667
"B6"    "JAX"   "EWR"   62.81818181818182
"B6"    "LAX"   "BZN"   60.0
"B6"    "JFK"   "ORD"   53.666666666666664
"B6"    "HPN"   "RSW"   52.35294117647059
"B6"    "PBI"   "PHL"   50.75
"B6"    "RSW"   "HPN"   50.05714285714286
"B6"    "PBI"   "LGA"   50.0
"B6"    "PHX"   "JFK"   49.111111111111114
"B6"    "EWR"   "BOS"   48.375
"B6"    "FLL"   "AUS"   44.9
"B6"    "SRQ"   "EWR"   44.666666666666664
"B6"    "LAX"   "MTJ"   44.0
"B6"    "JFK"   "PHX"   41.05555555555556
"B6"    "PHX"   "EWR"   40.5
"B6"    "LAX"   "RIC"   40.333333333333336
"B6"    "TPA"   "HPN"   40.1
"B6"    "LGA"   "PBI"   37.0
"B6"    "EWR"   "PHX"   36.666666666666664
"B6"    "PIT"   "FLL"   35.54545454545455
"B6"    "MTJ"   "LAX"   35.0
"B6"    "HPN"   "PBI"   34.63235294117647
"B6"    "RSW"   "PVD"   33.083333333333336
"B6"    "LGA"   "RSW"   33.0
"B6"    "PBI"   "DCA"   32.833333333333336
"B6"    "RSW"   "PHL"   31.3125
"B6"    "MSP"   "JFK"   30.454545454545453
"B6"    "SLC"   "BOS"   30.0
"B6"    "JFK"   "TPA"   29.52777777777778
"B6"    "FLL"   "JAX"   29.333333333333332
"B6"    "TPA"   "PHL"   29.11764705882353
"B6"    "AUS"   "FLL"   29.0
"B6"    "BZN"   "LAX"   29.0
```

“G4” Airline

"G4"	"GSO"	"SFB"	289.0
"G4"	"LAX"	"BOI"	252.4
"G4"	"LAS"	"EUG"	239.22222222222223
"G4"	"GSP"	"FLL"	220.85714285714286
"G4"	"FLL"	"GSP"	215.57142857142858
"G4"	"IAG"	"PGD"	214.0
"G4"	"BLI"	"OAK"	188.8
"G4"	"SWF"	"SFB"	187.0
"G4"	"MLI"	"PIE"	153.0
"G4"	"BLV"	"FLL"	151.0
"G4"	"FLL"	"BLV"	142.0
"G4"	"SFB"	"SWF"	139.0
"G4"	"ORF"	"FLL"	114.5
"G4"	"PGD"	"IAG"	101.5
"G4"	"HTS"	"PGD"	92.0
"G4"	"SPI"	"PGD"	81.1
"G4"	"PGD"	"HTS"	76.0
"G4"	"BNA"	"PGD"	75.0
"G4"	"FNT"	"SRQ"	74.4
"G4"	"SFB"	"BNA"	64.375
"G4"	"EWR"	"CVG"	64.0
"G4"	"GRR"	"LAS"	61.1
"G4"	"AZA"	"PVU"	60.93333333333333
"G4"	"MDT"	"AZA"	59.77777777777778
"G4"	"IDA"	"AZA"	59.6
"G4"	"CVG"	"CHS"	58.0
"G4"	"MDT"	"SRQ"	57.333333333333336
"G4"	"SRQ"	"SYR"	57.333333333333336
"G4"	"ABE"	"FLL"	57.0
"G4"	"FAR"	"BNA"	55.0
"G4"	"PGD"	"MEM"	53.0
"G4"	"CHS"	"CVG"	49.5
"G4"	"EWR"	"VPS"	48.0
"G4"	"IND"	"SRQ"	48.0
"G4"	"LAX"	"TUL"	46.75
"G4"	"FLL"	"SYR"	46.3
"G4"	"MEM"	"PGD"	46.0
"G4"	"OMA"	"AZA"	46.0
"G4"	"IND"	"PGD"	44.5
"G4"	"AZA"	"BLI"	44.1
"G4"	"FNT"	"PIE"	44.1

“MQ” Airline

"MQ"	"PIA"	"ORD"	152.33333333333334
"MQ"	"DFW"	"LAW"	62.58490566037736
"MQ"	"ORD"	"PBI"	48.25
"MQ"	"DCA"	"ORD"	47.0
"MQ"	"ORD"	"ABE"	45.666666666666664
"MQ"	"PBI"	"ORD"	45.25
"MQ"	"PNS"	"MIA"	45.25
"MQ"	"ORD"	"TOL"	44.0
"MQ"	"DAY"	"ORD"	42.34615384615385
"MQ"	"ICT"	"ORD"	41.70967741935484
"MQ"	"LBB"	"DFW"	40.0
"MQ"	"FAR"	"DFW"	38.47222222222222
"MQ"	"MIA"	"PNS"	38.0
"MQ"	"ABE"	"ORD"	37.666666666666664
"MQ"	"BZN"	"ORD"	36.37931034482759
"MQ"	"TUS"	"ORD"	36.36842105263158
"MQ"	"ORD"	"CMI"	33.92
"MQ"	"MSP"	"DFW"	31.244444444444444
"MQ"	"OKC"	"DFW"	29.878048780487806
"MQ"	"STL"	"ORD"	27.666666666666668
"MQ"	"ORD"	"BZN"	25.96551724137931
"MQ"	"ORF"	"ORD"	25.0
"MQ"	"MQT"	"ORD"	24.62962962962963
"MQ"	"TYS"	"DFW"	24.0
"MQ"	"SPI"	"DFW"	23.81132075471698
"MQ"	"MHK"	"DFW"	23.032258064516128
"MQ"	"DCA"	"DFW"	23.0
"MQ"	"RDU"	"DFW"	21.8
"MQ"	"ORD"	"SRQ"	21.742857142857144
"MQ"	"SGF"	"DFW"	21.73939393939394
"MQ"	"CWA"	"ORD"	21.413793103448278
"MQ"	"ORD"	"BOI"	21.333333333333332
"MQ"	"AMA"	"DFW"	21.234375
"MQ"	"ORD"	"EWR"	21.0
"MQ"	"CMI"	"DFW"	20.160714285714285
"MQ"	"BTR"	"DFW"	19.795454545454547
"MQ"	"ORD"	"AUS"	18.95
"MQ"	"SJT"	"DFW"	18.6484375
"MQ"	"CID"	"DFW"	18.473684210526315
"MQ"	"RIC"	"DFW"	18.339622641509433
"MQ"	"DFW"	"DCA"	18.333333333333332

3. [The number of times and reasons flights were canceled in Jan 2021] Cut-and-paste your mapper file, your reducer file, and a screenshot of the "cat" after your program has successfully executed on hadoop (e.g., "cat cancelations/part-00000 | more"). If you were unable to complete this part, show your mapper and your reducer and describe the problem (you were unable to debug).

Solution

The number of times and reasons flights were canceled in Jan 2021:

```
"A" 15985
"B" 54694
"C" 6440
"D"  6762
```

CancellationMapper.py

```
#!/usr/bin/env python3

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    line = line.split(",")
    # increase counters

    for code in line:
        if line[16] is not None:
                code = line[16]

        print ("{0}\t1".format(code))
```

CancellationReducer.py

```
#!/usr/bin/env python3

from operator import itemgetter
import sys

current_code = None
current_count = 0
code = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    code, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: code) before it is passed to the reducer
    if current_code == code:
        current_count += count
    else:
        if current_code:
            # write result to STDOUT
            print("{0}\t{1}".format(current_code, current_count))
        current_count = count
        current_code = code

# do not forget to output the last code if needed!
if current_code == code:
    print("{0}\t{1}".format(current_code, current_count))
```

cat cancelations/part-00000 | more

```
ubuntu@ip-172-31-67-92:~$ cat cancellations/part-00000 | more
""      8228963
"A"     15985
"B"     54694
"C"     6440
"D"     6762
```

4. [Comparing Jan 2021 to Jan 2020] How did each of the three results change when you looked at the Jan 2021 data? E.g., were flights canceled for the same reasons?

Task 1

Jan 2020	Jan 2021
1. "HA" 4.874697077690663	1. "HA" 2.5499843309307426
2. "WN" 5.17103650610911	2. "AS" 4.045389801538316
3. "DL" 7.048168351516365	3. "QX" 4.180262407301768
4. "YX" 7.738604038021231	4. "WN" 4.2302144024295645
5. "9E" 9.086734916140275	5. "F9" 5.304215419137682
6. "UA" 9.316666319581833	6. "UA" 6.034677215722582
7. "NK" 9.565280101540415	7. "YX" 6.278044909201251
8. "F9" 9.759827810266406	8. "9E" 6.334652752788773
9. "B6" 10.31781812274075	9. "NK" 6.558481377942105
10. "EV" 10.735447185813415	10. "DL" 6.782619542619543
11. "AA" 10.849758850964596	11. "AA" 8.18221389753601
12. "AS" 11.567723620266564	12. "OH" 9.439964011123834
13. "MQ" 13.045206454674894	13. "YV" 9.50642883013401
14. "OO" 13.915324959353104	14. "OO" 9.583526840595548
15. "G4" 14.494584366436122	15. "MQ" 10.104453441295547
16. "YV" 14.659875236078522	16. "G4" 12.029496923633731
17. "OH" 15.693528897037377	17. "B6" 12.907216494845361

In terms of on-time performance, HA still remained the top airline over 2020 and 2021. Airlines WN, DL, YX, and 9E performed within the top 2-5 respectively in 2020 wheareas 4 different airlines: AS, QX, WN, and F9 performed top 2-5 in 2021.

MQ, OO, YV, and G4 performed within the 5 worst airlines for on-time performance in both Jan 2020 and Jan 2021. OH, airline with worst performance in Jan 2020 was #12 in Jan 2021 and B6, airline with worst performance in Jan 2021 was #9 in Jan 2020.

However, on average Jan 2020 experienced higher delays on average: minimum = 4.87, maximum = 15.69, and average = 10.81. Jan 2021: minimum = 2.55, maximum = 12.91, average = 7.30

Task 2

Jan 2020	Jan 2021
1."B6" "RNO" "JFK" 44.3	1."B6" "PHL" "PBI" 191.0
2."B6" "HDN" "FLL" 43.22222222222222	2."B6" "LAX" "LAS" 93.0
3."B6" "ORH" "JFK" 35.096774193548384	3."B6" "LAS" "LAX" 87.5
4."B6" "SJU" "DCA" 34.806451612903224	4."B6" "SEA" "FLL" 68.0
5."B6" "ORH" "MCO" 32.903225806451616	5."B6" "ORD" "JFK" 64.66666666666667
6."B6" "BUR" "JFK" 32.666666666666664	6."B6" "JAX" "EWR" 62.81818181818182
7."B6" "STT" "SJU" 32.23076923076923	7."B6" "LAX" "BZN" 60.0
8."B6" "ORH" "FLL" 29.129032258064516	8."B6" "JFK" "ORD" 53.666666666666664
9."B6" "MCO" "SLC" 27.580645161290324	9."B6" "HPN" "RSW" 52.35294117647059
10. "B6""BOS" "SYR" 26.096774193548388	10. "B6""PBI" "PHL" 50.75
11. "B6""SJC" "BOS" 26.0	11. "B6""RSW" "HPN" 50.05714285714286
12. "B6""FLL" "ORD" 25.466666666666665	12. "B6""PBI" "LGA" 50.0
13. "B6""SLC" "MCO" 25.06451612903226	13. "B6""PHX" "JFK" 49.111111111111114
14. "B6""TPA" "EWR" 23.693548387096776	14. "B6""EWR" "BOS" 48.375
15. "B6""SJU" "STT" 23.23728813559322	15. "B6""FLL" "AUS" 44.9
1."G4" "GFK" "LAS" 202.33333333333334	1."G4" "GSO" "SFB" 289.0
2."G4" "LAS" "GFK" 185.22222222222223	2."G4" "LAX" "BOI" 252.4
3."G4" "CID" "LAS" 154.66666666666666	3."G4" "LAS" "EUG" 239.22222222222223
4."G4" "LCK" "SAV" 150.5	4."G4" "GSP" "FLL" 220.85714285714286
5."G4" "SAV" "LCK" 145.5	5."G4" "FLL" "GSP" 215.57142857142858
6."G4" "ABQ" "LAS" 129.0	6."G4" "IAG" "PGD" 214.0
7."G4" "MCI" "PGD" 126.55555555555556	7."G4" "BLI" "OAK" 188.8
8."G4" "SAV" "CVG" 110.44444444444444	8."G4" "SWF" "SFB" 187.0
9."G4" "SGF" "LAS" 109.55555555555556	9."G4" "MLI" "PIE" 153.0
10. "G4""TVC" "PIE" 101.375	10. "G4""BLV" "FLL" 151.0
11. "G4""AUS" "ABQ" 87.66666666666667	11. "G4""FLL" "BLV" 142.0
12. "G4""JAX" "ORF" 85.5	12. "G4""SFB" "SWF" 139.0
13. "G4""SCK" "AZA" 85.22222222222223	13. "G4""ORF" "FLL" 114.5
14. "G4""GRR" "MSY" 83.88888888888889	14. "G4""PGD" "IAG" 101.5
15. "G4""PVD" "PGD" 80.77777777777777	15. "G4""HTS" "PGD" 92.0
1."MQ" "ORD" "DCA" 66.25	1."MQ" "PIA" "ORD" 152.33333333333334
2."MQ" "SRQ" "DFW" 60.810810810810814	2."MQ" "DFW" "LAW" 62.58490566037736
3."MQ" "SDF" "LGA" 54.57142857142857	3."MQ" "ORD" "PBI" 48.25
4."MQ" "LBB" "DFW" 52.0	4."MQ" "DCA" "ORD" 47.0
5."MQ" "MQT" "ORD" 49.58620689655172	5."MQ" "ORD" "ABE" 45.666666666666664
6."MQ" "FAY" "CLT" 45.0	6."MQ" "PBI" "ORD" 45.25
7."MQ" "ORD" "SYR" 43.25	7."MQ" "PNS" "MIA" 45.25
8."MQ" "EYW" "DFW" 41.44444444444444	8."MQ" "ORD" "TOL" 44.0
9."MQ" "RIC" "JFK" 39.43333333333333	9."MQ" "DAY" "ORD" 42.34615384615385
10. "MQ""JFK" "PHL" 36.125	10. "MQ""ICT" "ORD" 41.70967741935484
11. "MQ""CLE" "MIA" 35.28	11. "MQ""LBB" "DFW" 40.0
12. "MQ""SAT" "ORD" 35.06666666666667	12. "MQ""FAR" "DFW" 38.47222222222222
13. "MQ""GCK" "DFW" 34.517857142857146	13. "MQ""MIA" "PNS" 38.0
14. "MQ""CVG" "JFK" 32.064516129032256	14. "MQ""ABE" "ORD" 37.666666666666664
15. "MQ""DFW" "GCK" 31.228070175438596	15. "MQ""BZN" "ORD" 36.37931034482759

The three worst flights in on-time performance in January 2021 (B6, G4, MQ) were compared to the same flights in January 2020 - however this does not mean these flights were the worst performing in Jan 2020. The average delay time for 15 of the worst routes for each plane was calculated. For all three flights, the worst 15 routes of January 2021 had no route commonalities between the airlines' worst 15 routes of January 2020.

Task 3

"A"="Carrier", "B"="Weather", "C"="National Air System", and "D"="Security"

Jan 2020	Jan 2021
"A" 31004	"A" 15985
"B" 114747	"B" 54694
"C" 13547	"C" 6440
"D" 46	"D" 6762

In Jan 2020, there were a total of 159,344 cancellations. In Jan 2021, there were a total of 83,881 cancellations. This indicates that January 2020 experienced about 90% more cancellations than January 2021.

There were about 94% more cancellations labeled "A" (Carrier) in Jan 2020 than that of Jan 2021, 109% more cancellations labeled "B" (Weather) in Jan 2020 than that of Jan 2021, 110% more cancellations labeled "C" (National) in Jan 2020 than that of Jan 2021, and 0.7% of cancellations labeled "D" (Security) for Jan 2021 compared to that of Jan 2021.

Conclusion

From evidence above, it can be concluded that January 2020 experienced higher delays than that of January 2021. January 2020 experienced a higher average in flight delays across all airlines listed – even when 4 out of the 5 worst performing airlines in January 2020 remained the same as that of January 2021. When analyzing the 15 worst routes for the three worst airlines in terms of delay, it can be concluded that the route taken did not have influence for the airline's delay performance between the years – in other words, there were no common routes between the two years. Finally, January 2020 experienced higher number of cancellations than that of January 2021. For cancellations labeled – A, B, and C, there were about 100% more cancellations for these reasons in 2020 than that of 2021. In both 2020 and 2021, flights were canceled for similar reasons (mostly due to weather, followed by carrier cancelations). January 2021 experienced a much higher proportion of security cancellations than that of January 2021.