Storm Impacts on Aviation

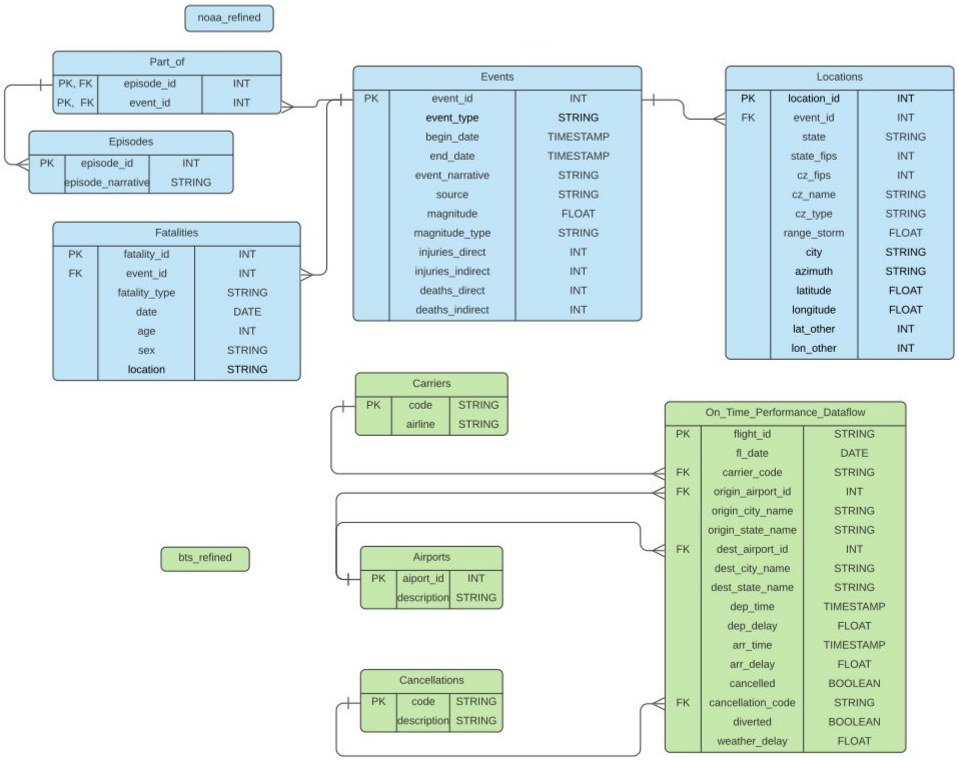
Kathy Wang, Sierra Obermoeller-Gilmer

Areas of Interest

- What storm events have the greatest effects on aviation?
- In what ways are different airlines affected by storm events?
- Which airports are most affected by storm related delays?

The Datasets

- NOAA Storm Events Dataset
- Bureau of Transportation Statistics Airline On-Time Performance Dataset



noaa_refined

Part_of

PK, FK	episode_id	INT
PK, FK	event_id	INT

Episodes

PK	episode_id	INT
	episode_narrative	STRING

Fatalities

PK	fatality_id	INT
FK	event_id	INT
	fatality_type	STRING
	date	DATE
	age	INT
	sex	STRING
	location	STRING

Events

PK	event_id	INT
	event_type	STRING
	begin_date	TIMESTAMP
	end_date	TIMESTAMP
	event_narrative	STRING
	source	STRING
	magnitude	FLOAT
	magnitude_type	STRING
	injuries_direct	INT
	injuries_indirect	INT
	deaths_direct	INT
	deaths_indirect	INT

Locations

PK	location_id	INT
FK	event_id	INT
	state	STRING
	state_fips	INT
	cz_fips	INT
	cz_name	STRING
	cz_type	STRING
	range_storm	FLOAT
	city	STRING
	azimuth	STRING
	latitude	FLOAT
	longitude	FLOAT
	lat_other	INT
	lon_other	INT

bts_refined

Carriers

PK	code	STRING
	airline	STRING

Airports

PK	aiport_id	INT
	description	STRING

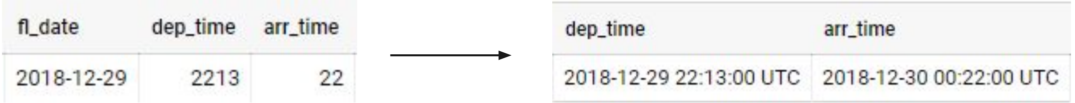
Cancellations

PK	code	STRING
	description	STRING

On_Time_Performance_Dataflow

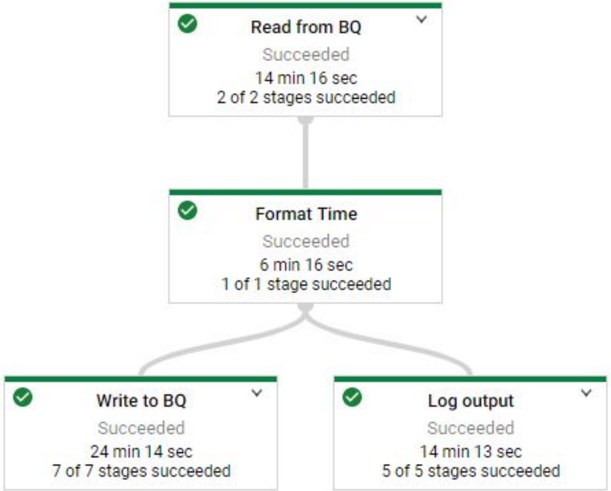
PK	flight_id	STRING
	fl_date	DATE
FK	carrier_code	STRING
FK	origin_airport_id	INT
	origin_city_name	STRING
	origin_state_name	STRING
FK	dest_airport_id	INT
	dest_city_name	STRING
	dest_state_name	STRING
	dep_time	TIMESTAMP
	dep_delay	FLOAT
	arr_time	TIMESTAMP
	arr_delay	FLOAT
	cancelled	BOOLEAN
FK	cancellation_code	STRING
	diverted	BOOLEAN
	weather_delay	FLOAT

Beam Pipeline



fl_date	dep_time	arr_time
2018-12-29	2213	22

dep_time	arr_time
2018-12-29 22:13:00 UTC	2018-12-30 00:22:00 UTC

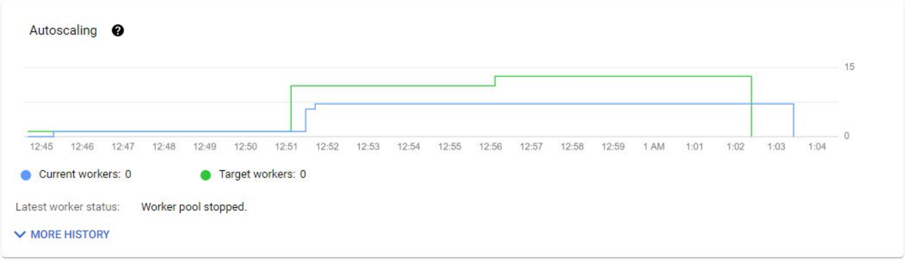


Read from BQ
Succeeded
14 min 16 sec
2 of 2 stages succeeded

Format Time
Succeeded
6 min 16 sec
1 of 1 stage succeeded

Write to BQ
Succeeded
24 min 14 sec
7 of 7 stages succeeded

Log output
Succeeded
14 min 13 sec
5 of 5 stages succeeded

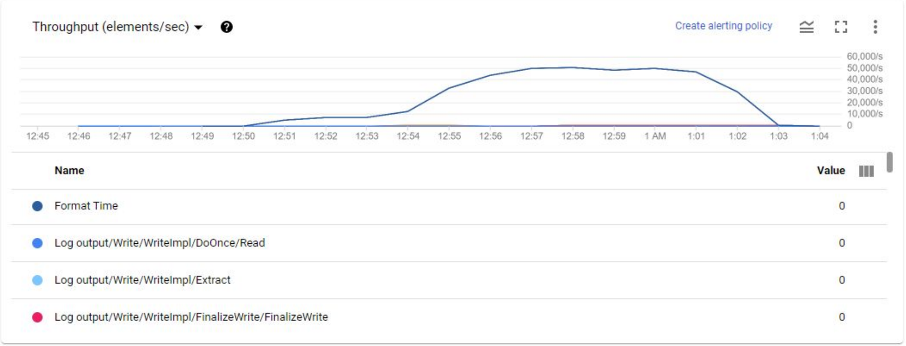


Autoscaling

● Current workers: 0 ● Target workers: 0

Latest worker status: Worker pool stopped.

MORE HISTORY



Throughput (elements/sec)

Name	Value
● Format Time	0
● Log output/Write/WriteImpl/DoOnce/Read	0
● Log output/Write/WriteImpl/Extract	0
● Log output/Write/WriteImpl/FinalizeWrite/FinalizeWrite	0

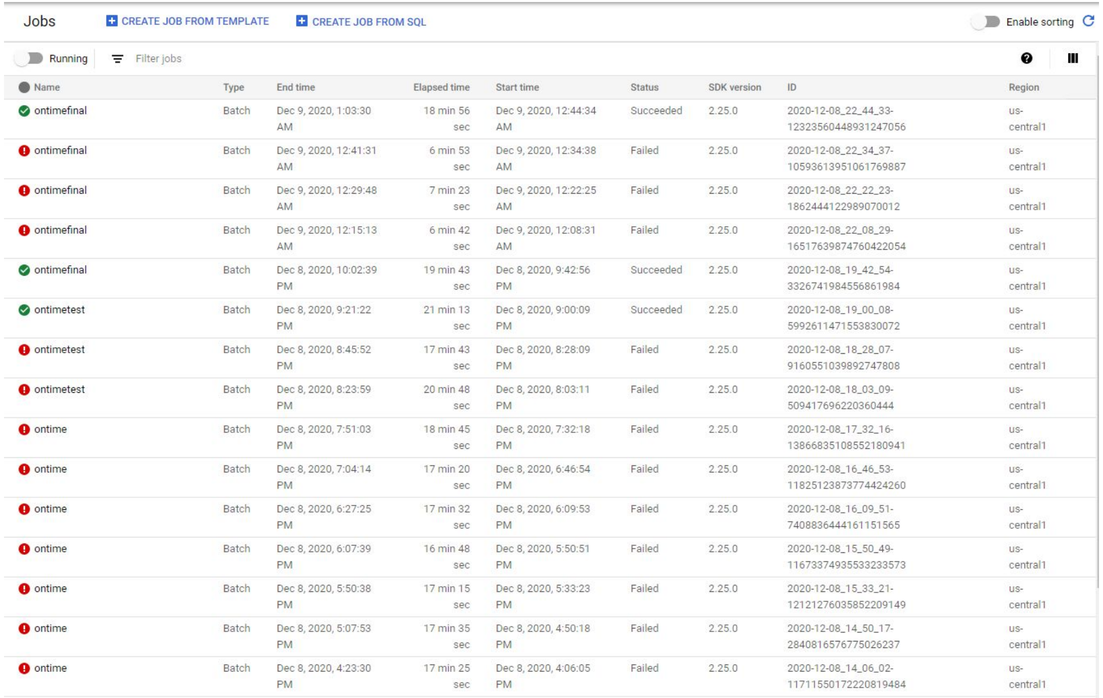
Conversion of Time

```
class FormatTime(beam.DoFn):
 def process(self, element):
   from datetime import datetime, timedelta
   import datetime
#checks if arrival time is the next day, and if it is sets the arrival
date 1 day after flight
   arr_date= fl_date
   if dep_time is not None and arr_time is not None and dep_time!=''
and arr_time!='' and fl_date is not None and fl_date!='':
       if int(dep_time)> int(arr_time) or int(arr_time)==2400:
           temp_date=
datetime.datetime.strptime(str(fl_date),"%Y-%m-%d")
           arr_date=temp_date + timedelta(days=1)
           array_date= str(arr_date).split(" ")
           arr_date=array_date[0]
       else:
           arr_date=fl_date
```

```
# Handle time conversion
   # Convert departure time from int to time
   # New format will be hh:mm:ss
   dep_time_new = dep_time
   dep_time = str(dep_time)
   if dep_time != 'None' and dep_time != '':
       if dep_time == '2400':
           dep_time_new = '00:00:00'
       elif len(dep_time) == 3:
           dep_time_new ='0' + dep_time[:1] + ':' +
dep_time[1:] + ':00'
       elif len(dep_time) == 4:
           dep_time_new = dep_time[:2] + ':' + dep_time[2:] +
':00'
       elif len(dep_time) == 2:
           dep_time_new ='00:' + dep_time + ':00'
       elif len(dep_time) == 1:
           dep_time_new ='00:0' + dep_time + ':00'
       else:
           dep_time_new = '00:00:00'
   else:
       dep_time_new = '00:00:00'
```

Challenges

- > 20,000,000 entities
- Formatting time fields for cross dataset joins
- Debugging data flow errors
    - 24:00:00 does not work in TIME
    - Decision to change to TIMESTAMP

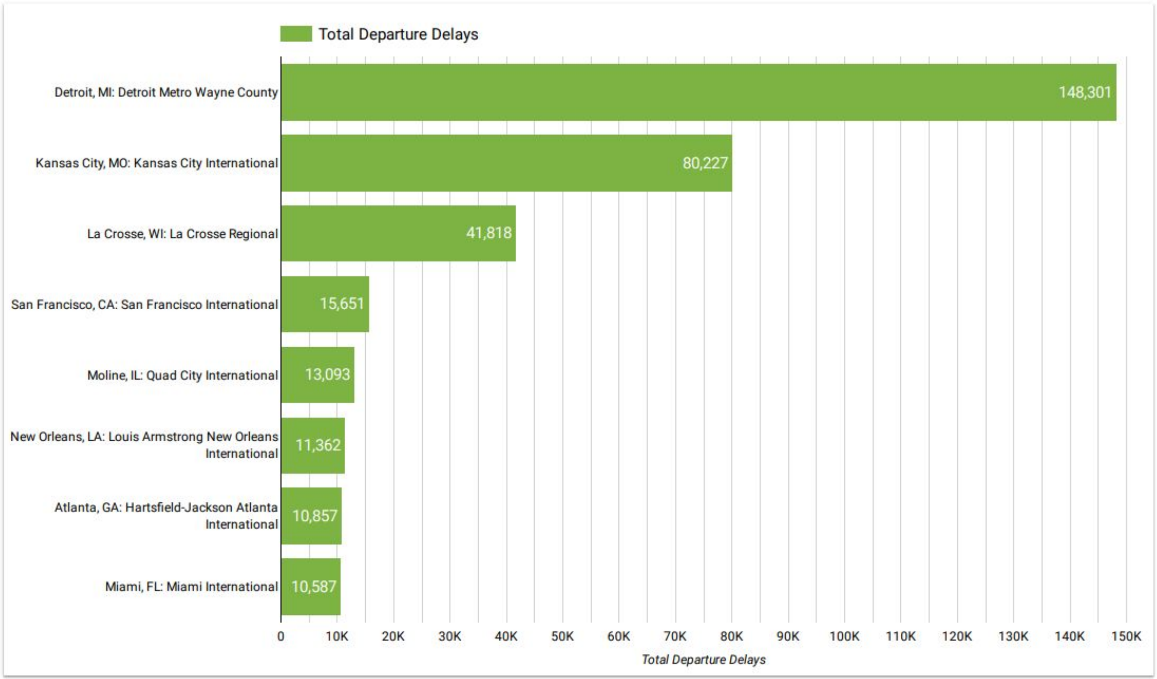


Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region
✓ ontimefinal	Batch	Dec 9, 2020, 1:03:30 AM	18 min 56 sec	Dec 9, 2020, 12:44:34 AM	Succeeded	2.25.0	2020-12-08_22_44_33-12323560448931247056	us-central1
! ontimefinal	Batch	Dec 9, 2020, 12:41:31 AM	6 min 53 sec	Dec 9, 2020, 12:34:38 AM	Failed	2.25.0	2020-12-08_22_34_37-10593613951061769887	us-central1
! ontimefinal	Batch	Dec 9, 2020, 12:29:48 AM	7 min 23 sec	Dec 9, 2020, 12:22:25 AM	Failed	2.25.0	2020-12-08_22_22_23-1862444122989070012	us-central1
! ontimefinal	Batch	Dec 9, 2020, 12:15:13 AM	6 min 42 sec	Dec 9, 2020, 12:08:31 AM	Failed	2.25.0	2020-12-08_22_08_29-16517639874760422054	us-central1
✓ ontimefinal	Batch	Dec 8, 2020, 10:02:39 PM	19 min 43 sec	Dec 8, 2020, 9:42:56 PM	Succeeded	2.25.0	2020-12-08_19_42_54-3326741984556861984	us-central1
✓ ontimetest	Batch	Dec 8, 2020, 9:21:22 PM	21 min 13 sec	Dec 8, 2020, 9:00:09 PM	Succeeded	2.25.0	2020-12-08_19_00_08-5992611471553830072	us-central1
! ontimetest	Batch	Dec 8, 2020, 8:45:52 PM	17 min 43 sec	Dec 8, 2020, 8:28:09 PM	Failed	2.25.0	2020-12-08_18_28_07-9160551039892747808	us-central1
! ontimetest	Batch	Dec 8, 2020, 8:23:59 PM	20 min 48 sec	Dec 8, 2020, 8:03:11 PM	Failed	2.25.0	2020-12-08_18_03_09-509417696220360444	us-central1
! ontime	Batch	Dec 8, 2020, 7:51:03 PM	18 min 45 sec	Dec 8, 2020, 7:32:18 PM	Failed	2.25.0	2020-12-08_17_32_16-13866835108552180941	us-central1
! ontime	Batch	Dec 8, 2020, 7:04:14 PM	17 min 20 sec	Dec 8, 2020, 6:46:54 PM	Failed	2.25.0	2020-12-08_16_46_53-11825123873774424260	us-central1
! ontime	Batch	Dec 8, 2020, 6:27:25 PM	17 min 32 sec	Dec 8, 2020, 6:09:53 PM	Failed	2.25.0	2020-12-08_16_09_51-7408836444161151565	us-central1
! ontime	Batch	Dec 8, 2020, 6:07:39 PM	16 min 48 sec	Dec 8, 2020, 5:50:51 PM	Failed	2.25.0	2020-12-08_15_50_49-11673374935533233573	us-central1
! ontime	Batch	Dec 8, 2020, 5:50:38 PM	17 min 15 sec	Dec 8, 2020, 5:33:23 PM	Failed	2.25.0	2020-12-08_15_33_21-12121276035852209149	us-central1
! ontime	Batch	Dec 8, 2020, 5:07:53 PM	17 min 35 sec	Dec 8, 2020, 4:50:18 PM	Failed	2.25.0	2020-12-08_14_50_17-2840816576775026237	us-central1
! ontime	Batch	Dec 8, 2020, 4:23:30 PM	17 min 25 sec	Dec 8, 2020, 4:06:05 PM	Failed	2.25.0	2020-12-08_14_06_02-11711550172220819484	us-central1

Query 1

```
SELECT a.description,  SUM(a.dep_delay) AS
total_departure_delays
FROM (SELECT  * FROM (SELECT * FROM
                    bts_refined.On_Time_Performance_Dataflow
                    WHERE EXTRACT(YEAR FROM fl_date) =
          2019) AS o
          INNER JOIN bts_refined.Airports c
          ON c.airport_id = o.origin_airport_id) AS a
INNER JOIN (SELECT  *
          FROM noaa_refined.Events a
          INNER JOIN noaa_refined.Locations l
          ON l.event_id = a.event_id
          WHERE extract(year from a.begin_date) = 2019) AS b
ON UPPER(a.origin_state_name) = b.state
          AND UPPER(a.origin_city_name) = b.city
          AND a.dep_time BETWEEN b.begin_date AND
b.end_date
WHERE a.dep_delay > 0
GROUP BY a.description
HAVING SUM(a.dep_delay) > 10000
ORDER BY SUM(a.dep_delay) DESC LIMIT 8;
```

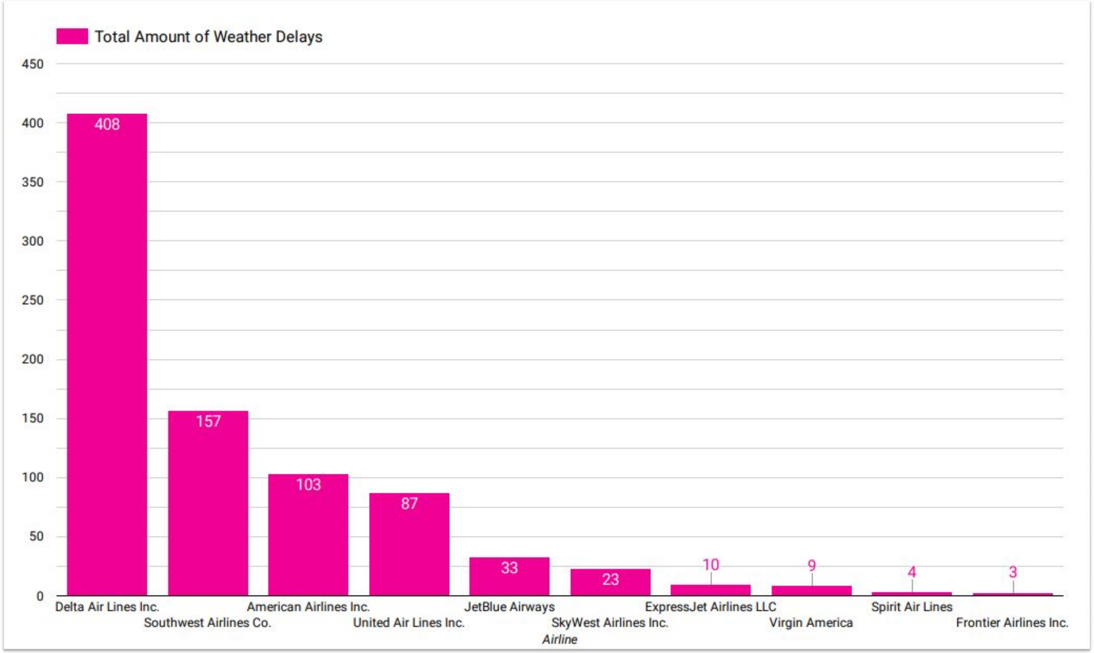
Airports with the Longest Storm Related Departure Delays in 2019



Query 2

```
SELECT a.airline, COUNT(*) AS count
FROM (SELECT * FROM (SELECT * FROM
                    bts_refined.On_Time_Performance_Dataflow
                    WHERE EXTRACT(YEAR FROM fl_date) =
          2017) AS o
          INNER JOIN bts_refined.Carriers c
          ON c.code = o.carrier_code) AS a
INNER JOIN (SELECT *
          FROM noaa_refined.Events a
          INNER JOIN noaa_refined.Locations l
          ON l.event_id = a.event_id
          WHERE EXTRACT(YEAR FROM a.begin_date) = 2017)
AS b ON UPPER(a.origin_state_name) = b.state
          AND UPPER(a.origin_city_name) = b.city
          AND a.dep_time BETWEEN b.begin_date AND
b.end_date
WHERE a.weather_delay > 0
GROUP BY a.airline
ORDER BY COUNT(*) desc
LIMIT 10;
```

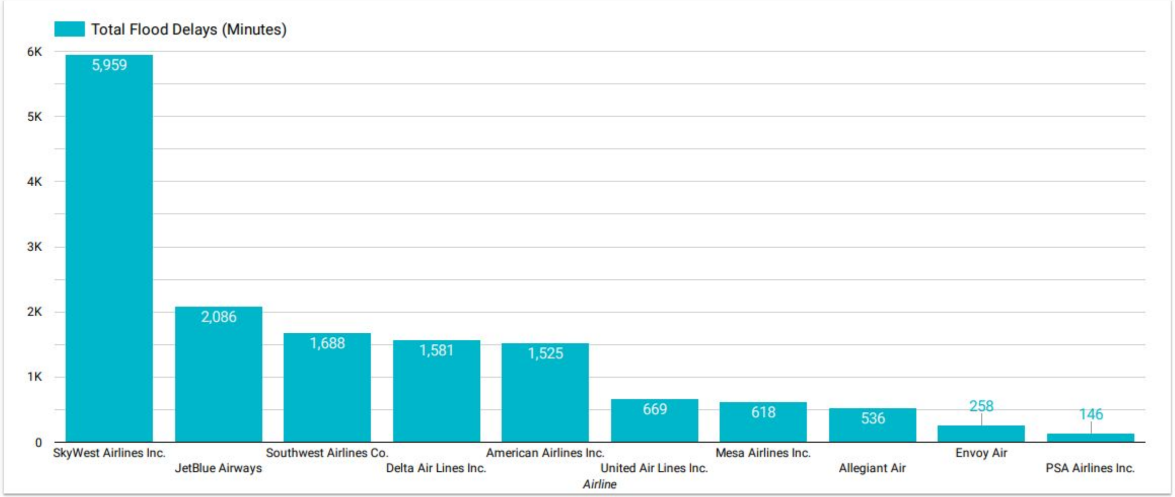
Airlines with the Most Storm Related Weather Delays in 2017



Query 3

```
SELECT a.airline, SUM(a.dep_delay) AS total_flood_delays
FROM (SELECT * FROM (SELECT * from
                        bts_refined.On_Time_Performance_Dataflow
                        WHERE EXTRACT(YEAR FROM fl_date) =
            2018) AS o
            INNER JOIN bts_refined.Carriers c
            ON c.code = o.carrier_code) AS a
INNER JOIN (SELECT  *
            FROM noaa_refined.Events a
            INNER JOIN noaa_refined.Locations l
            ON l.event_id = a.event_id
            WHERE EXTRACT(YEAR FROM a.begin_date) = 2018
            AND a.event_type = 'Flood') AS b
ON UPPER(a.origin_state_name) = b.state
            AND UPPER(a.origin_city_name) = b.city
            AND a.dep_time BETWEEN b.begin_date AND
b.end_date
WHERE a.weather_delay > 0
GROUP BY a.airline
ORDER BY SUM(a.dep_delay) desc
LIMIT 10;
```

Airlines with the Longest Total Flood Related Weather Delays in 2018



Future Improvements

- Formatting airport location for more accurate cross-dataset joins
    - Get latitude and longitude for airport
- Flights that were cancelled by storms at the origin vs destination
- Use NTSB Aviation Accidents dataset