Project

Weather and Airline Delay Analysis

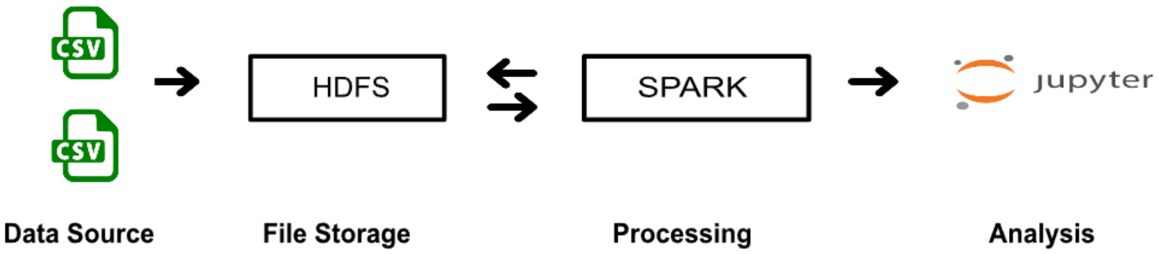
BUAN 6346.SW1 – Big Data

Submitted By- Sungho Yim

Aditya Chouksey

In this report we aim to assess the effects of various weather conditions on flight delays.

The architecture that we decided to use is Hadoop and using Spark queries on Jupyter Notebook.



In order to answer the business questions related to weather and airline delay, we acquired data from Kaggle.

Information about Data and their Sources:

Weather: US Weather Events)| US Weather Events (2016 - 2021) | Kaggle

Description:

This is a countrywide weather events dataset that includes 7.5 million events and covers 49 states of the United States. Examples of weather events are *rain*, *snow*, *storm*, and *freezing condition*.

Some of the events in this dataset are extreme events (e.g., storm) and some could be regarded as regular events (e.g. rain and snow). The data is collected from January 2016 to December 2021, using historical weather reports that were collected from 2,071 airport-based weather stations across the nation.

Description of Weather Events

Weather event is a spatiotemporal entity, where such an entity is associated with location and time. Following is the description of available weather event types in dataset:

- Severe-Cold: The case of having extremely low temperature, with temperature below -23.7 degrees of Celsius.
- Fog: The case where there is low visibility condition as a result of fog or haze.
- Hail: The case of having solid precipitation including ice pellets and hail.
- Rain: The case of having rain, ranging from light to heavy.
- Snow: The case of having snow, ranging from light to heavy.
- Storm: The extremely windy condition, where the wind speed is **at least 60 km/h**.
- Other Precipitation: Any other type of precipitation which cannot be assigned to previously described event types.

Airline Delay and Cancellation Data

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flight.
The datasets contain daily airline information covering from flight information, carrier company, to taxing-in, taxing-out time, and generalized delay reason of exactly 10 years.

Airline Delay and Cancellation Data, 2009 - 2018 | Kaggle

Business Questions:

We are interested in Departure and Arrival delay due to weather and the cancellation of flights subsequently from the two datasets. We are analyzing the data from the year 2016-17.

Is there a correlation between hail and flight delays?

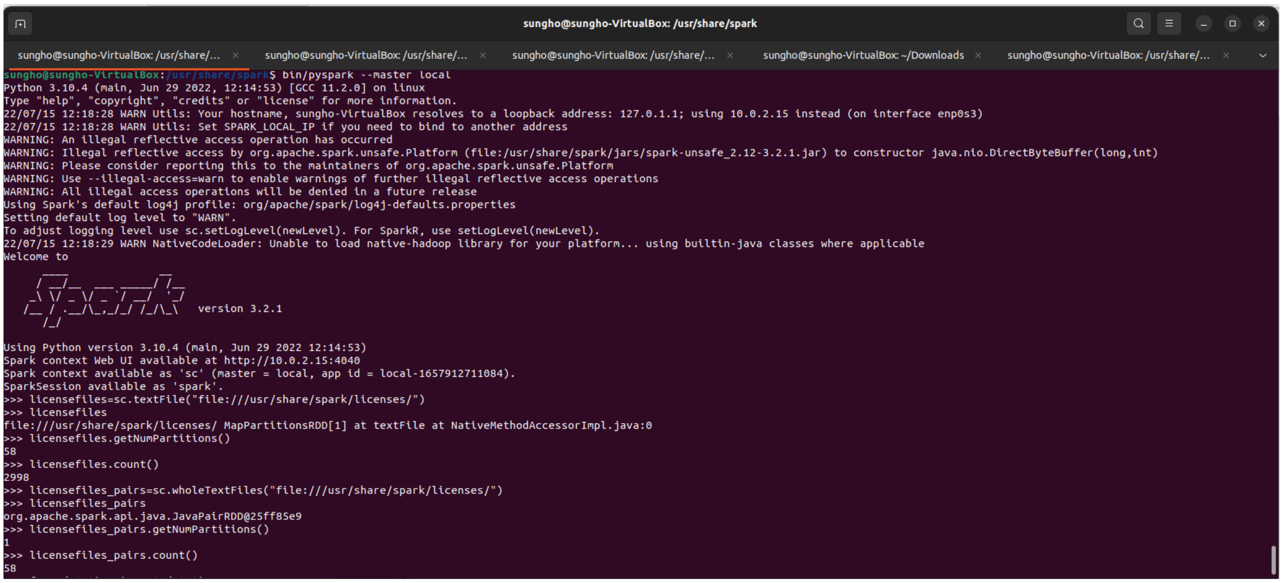
Is there a correlation between cold weather days and flight delays?

Which weather conditions have the most impact on flight delays?

Frequency of weather events (fog, storm) affecting the airline-time

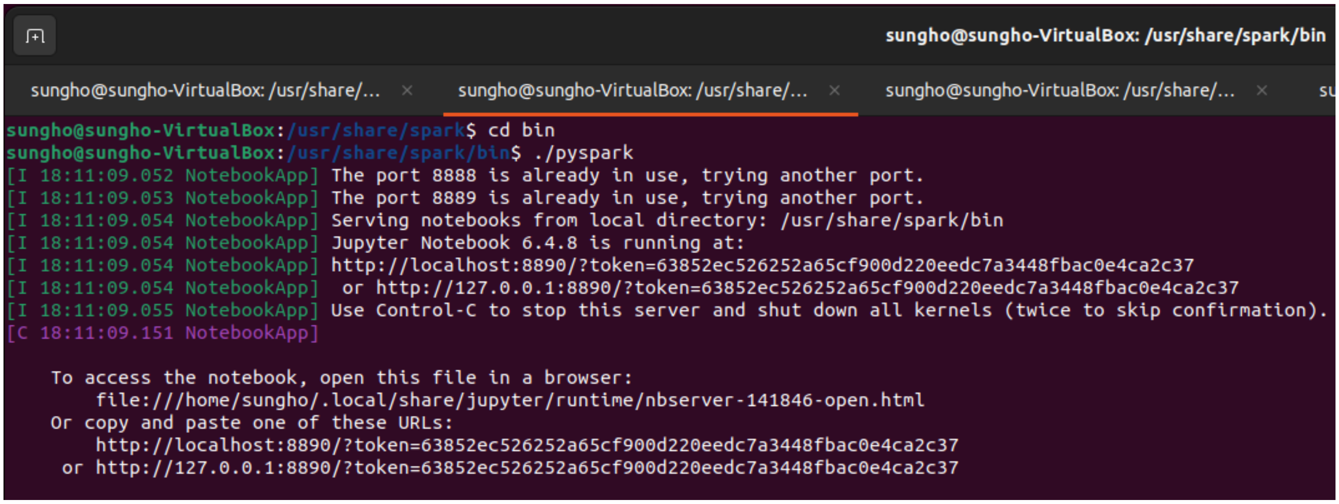
The percentage change of weather events from year 2016 to 2017, affecting the airline time

Following are the queries and analysis:


sungho@sungho-VirtualBox: /usr/share/spark
sungho@sungho-VirtualBox: /usr/share/...
sungho@sungho-VirtualBox: /usr/share/...
sungho@sungho-VirtualBox: /usr/share/...
sungho@sungho-VirtualBox: ~/Downloads
sungho@sungho-VirtualBox: /usr/share/...

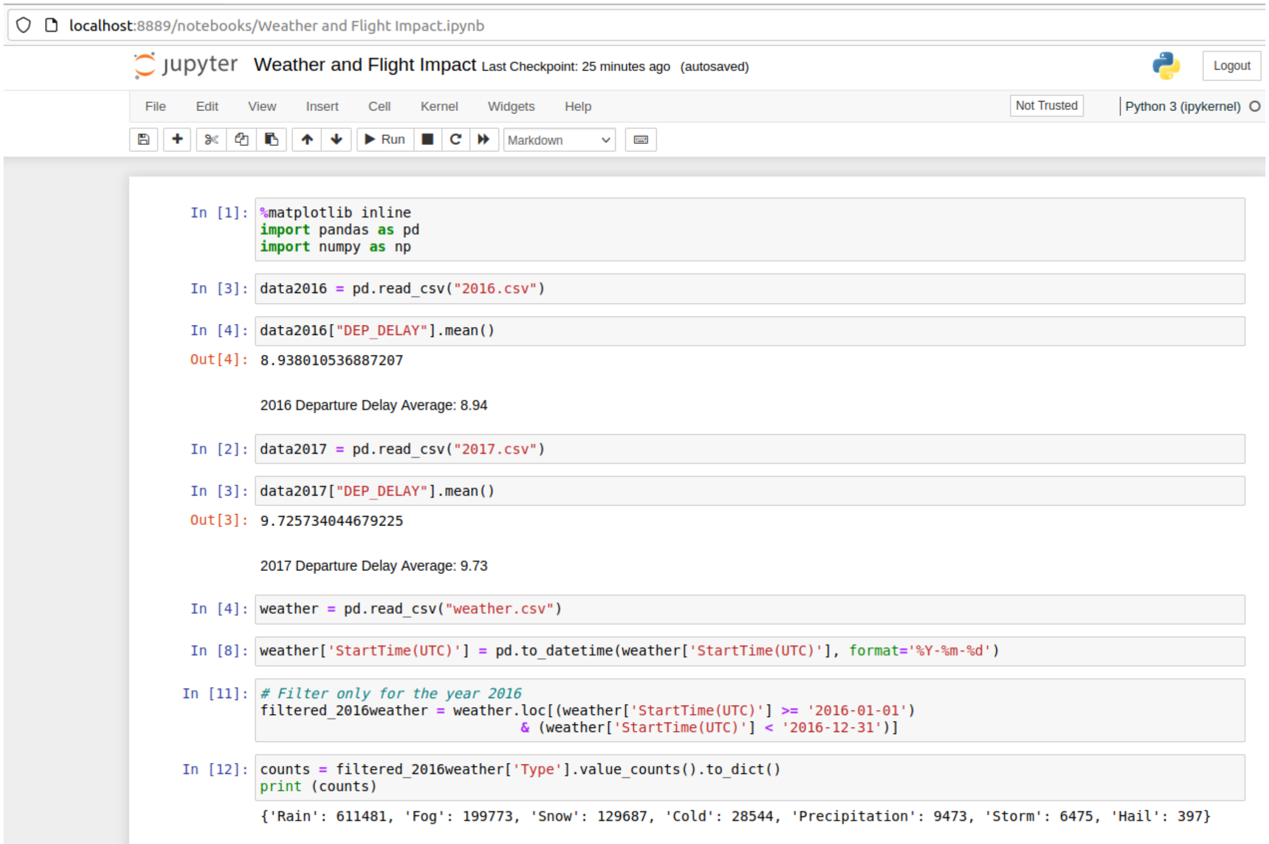
```
sungho@sungho-VirtualBox:/usr/share/spark$ bin/pyspark --master local
Python 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
22/07/15 12:18:28 WARN Utils: Your hostname, sungho-VirtualBox resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
22/07/15 12:18:28 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/usr/share/spark/jars/spark-unsafe_2.12-3.2.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/07/15 12:18:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Python version 3.10.4 (main, Jun 29 2022 12:14:53)
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local, app id = local-1657912711084).
SparkSession available as 'spark'.
>>> licensefiles=sc.textFile("file:///usr/share/spark/licenses/")
>>> licensefiles
file:///usr/share/spark/licenses/ MapPartitionsRDD[1] at textFile at NativeMethodAccessorImpl.java:0
>>> licensefiles.getNumPartitions()
58
>>> licensefiles.count()
2998
>>> licensefiles_pairs=sc.wholeTextFiles("file:///usr/share/spark/licenses/")
>>> licensefiles_pairs
org.apache.spark.api.java.JavaPairRDD@25ff85e9
>>> licensefiles_pairs.getNumPartitions()
1
>>> licensefiles_pairs.count()
58
```


sungho@sungho-VirtualBox: /usr/share/spark/bin
sungho@sungho-VirtualBox: /usr/share/...
sungho@sungho-VirtualBox: /usr/share/...
sungho@sungho-VirtualBox: /usr/share/...
su

```
sungho@sungho-VirtualBox:/usr/share/spark$ cd bin
sungho@sungho-VirtualBox:/usr/share/spark/bin$ ./pyspark
[I 18:11:09.052 NotebookApp] The port 8888 is already in use, trying another port.
[I 18:11:09.053 NotebookApp] The port 8889 is already in use, trying another port.
[I 18:11:09.054 NotebookApp] Serving notebooks from local directory: /usr/share/spark/bin
[I 18:11:09.054 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 18:11:09.054 NotebookApp] http://localhost:8890/?token=63852ec526252a65cf900d220eedc7a3448fbac0e4ca2c37
[I 18:11:09.054 NotebookApp]  or http://127.0.0.1:8890/?token=63852ec526252a65cf900d220eedc7a3448fbac0e4ca2c37
[I 18:11:09.055 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:11:09.151 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/sungho/.local/share/jupyter/runtime/nbserver-141846-open.html
    Or copy and paste one of these URLs:
        http://localhost:8890/?token=63852ec526252a65cf900d220eedc7a3448fbac0e4ca2c37
     or http://127.0.0.1:8890/?token=63852ec526252a65cf900d220eedc7a3448fbac0e4ca2c37
```


localhost:8889/notebooks/Weather and Flight Impact.ipynb
jupyter
Weather and Flight Impact Last Checkpoint: 25 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown

```
In [1]: %matplotlib inline
        import pandas as pd
        import numpy as np

In [3]: data2016 = pd.read_csv("2016.csv")

In [4]: data2016["DEP_DELAY"].mean()

Out[4]: 8.938010536887207
```

2016 Departure Delay Average: 8.94

```
In [2]: data2017 = pd.read_csv("2017.csv")

In [3]: data2017["DEP_DELAY"].mean()

Out[3]: 9.725734044679225
```

2017 Departure Delay Average: 9.73

```
In [4]: weather = pd.read_csv("weather.csv")

In [8]: weather['StartTime(UTC)'] = pd.to_datetime(weather['StartTime(UTC)'], format='%Y-%m-%d')

In [11]: # Filter only for the year 2016
         filtered_2016weather = weather.loc[(weather['StartTime(UTC)'] >= '2016-01-01')
                                            & (weather['StartTime(UTC)'] < '2016-12-31')]

In [12]: counts = filtered_2016weather['Type'].value_counts().to_dict()
         print (counts)

         {'Rain': 611481, 'Fog': 199773, 'Snow': 129687, 'Cold': 28544, 'Precipitation': 9473, 'Storm': 6475, 'Hail': 397}
```

2016 Data Frequency:

Fog: 199,773

Storm: 6475

Rain: 611,481

Snow: 129,687

Cold: 28,544

Precipitation: 9473

Hail: 397

In [13]:

```
# Filter only for the year 2017
filtered_2017weather = weather.loc[(weather['StartTime(UTC)'] >= '2017-01-01')
                              & (weather['StartTime(UTC)'] < '2017-12-31')]
```

In [14]:

```
counts2 = filtered_2017weather['Type'].value_counts().to_dict()
print (counts2)
```

```
{'Rain': 653696, 'Fog': 204196, 'Snow': 122452, 'Cold': 26660, 'Precipitation': 9467, 'Storm': 6747, 'Hail': 476}
```

2017 Data Frequency:

Fog: 204,196

Storm: 6747

Rain: 653,696

Snow: 122,452

Cold: 26,660

Precipitation: 9467

Hail: 476

Comparison from 2016-2017:

2016 Departure Delay Average: 8.94 Fog: 199,773 Storm: 6475

2017 Departure Delay Average: 9.73 Fog: 204,196 Storm: 6747

In [15]:

```python
# % increase in Storms:
storm_inc = (6747-6475)/6475
print (storm_inc)
```

```
0.04200772200772201
```

% Increase in Storms: 4.20%

In [16]:

```python
# % increase in Fogs:
fog_inc = (204196-199773)/199773
print (fog_inc)
```

```
0.02214012904646774
```

% Increase in Fogs: 2.21%

In [29]:

```python
# % increase in Snows:
snow_inc = (122452-129687)/129687
print (snow_inc)
```

```
-0.055788166894137424
```

% Decrease in Snow: 5.58%

In [30]:

```python
# % increase in Rains:
rain_inc = (653696-611481)/611481
print (rain_inc)
```

```
0.06903730451150568
```

% Increase in Rains: 6.90%

In [31]:

```python
# % increase in Hail:
hail_inc = (476-397)/397
print (hail_inc)
```

```
0.19899244332493704
```

% Increase in Hail: 19.9%

In [34]:

```python
# % increase in Cold:
cold_inc = (26660-28544)/28544
print (cold_inc)
```

```
-0.06600336322869955
```

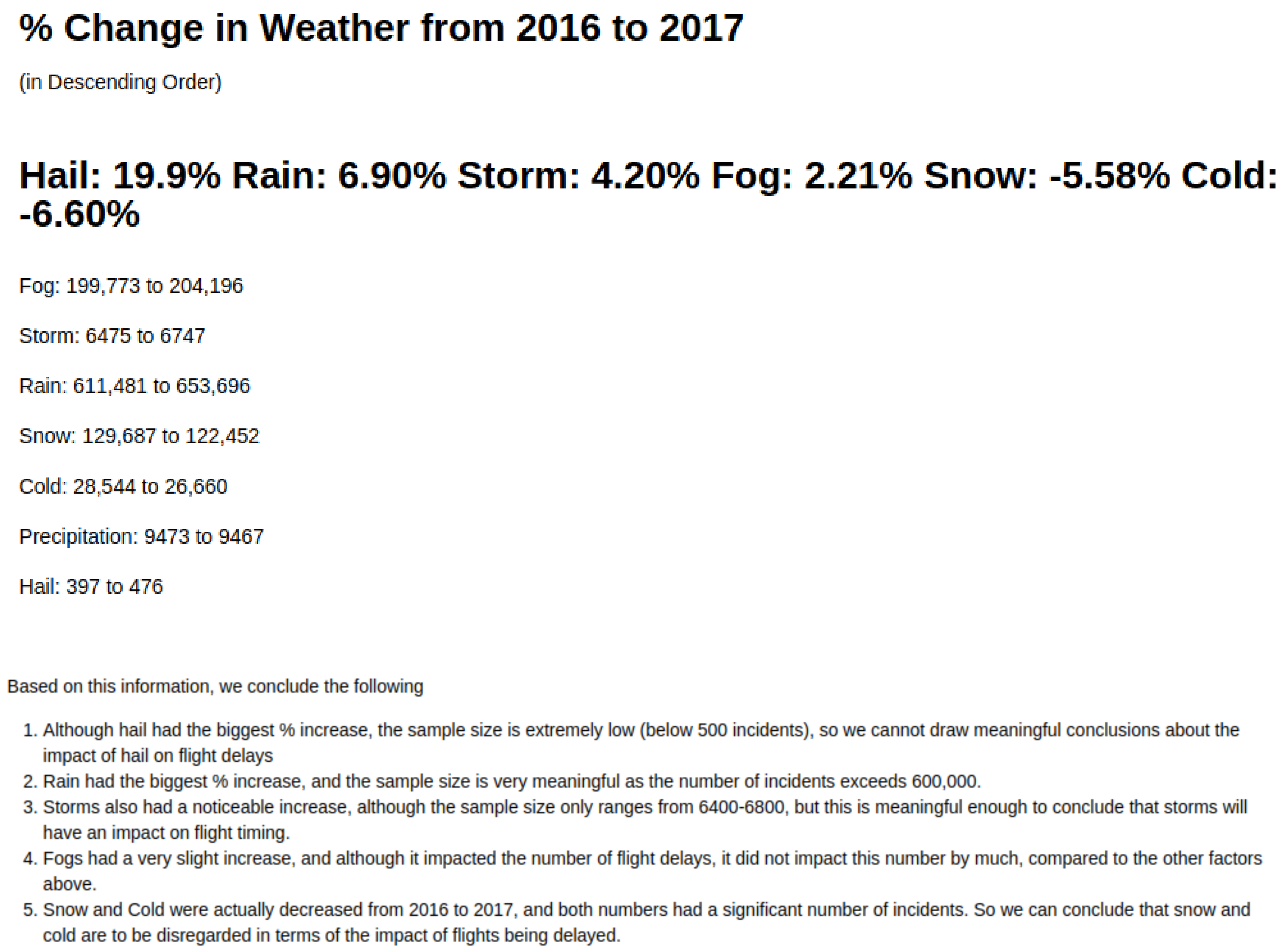
% Decrese in Cold: 6.60%

In [17]:

```python
# % increase in Departure Delays:
dep_inc = (9.73-8.94)/8.94
print (dep_inc)
```

```
0.08836689038031331
```

% Increase in Departure Delay Time from 2016 to 2017: 8.84%



% Change in Weather from 2016 to 2017

(in Descending Order)

Hail: 19.9% Rain: 6.90% Storm: 4.20% Fog: 2.21% Snow: -5.58% Cold: -6.60%

Fog: 199,773 to 204,196

Storm: 6475 to 6747

Rain: 611,481 to 653,696

Snow: 129,687 to 122,452

Cold: 28,544 to 26,660

Precipitation: 9473 to 9467

Hail: 397 to 476

Based on this information, we conclude the following

1. Although hail had the biggest % increase, the sample size is extremely low (below 500 incidents), so we cannot draw meaningful conclusions about the impact of hail on flight delays
2. Rain had the biggest % increase, and the sample size is very meaningful as the number of incidents exceeds 600,000.
3. Storms also had a noticeable increase, although the sample size only ranges from 6400-6800, but this is meaningful enough to conclude that storms will have an impact on flight timing.
4. Fogs had a very slight increase, and although it impacted the number of flight delays, it did not impact this number by much, compared to the other factors above.
5. Snow and Cold were actually decreased from 2016 to 2017, and both numbers had a significant number of incidents. So we can conclude that snow and cold are to be disregarded in terms of the impact of flights being delayed.

Final answers to the business questions:

Is there a correlation between hail and flight delays?

Although hail had the biggest percentage increase from 2016 to 2017, the sample size is extremely low (below 500 incidents). Therefore, we cannot draw any real meaningful conclusions about the impact of hail on flight delays, so there is no correlation between hail and flight delays.

Is there a correlation between cold weather days and flight delays?

The number of cold weather days has actually seen a decrease in frequency from 2016 to 2017, as the frequency dropped from 28,544 to 26,660, which is a 6.6% decrease. Based on this information, the conclusion is that there is no correlation between cold weather days and flight delays.

Which weather conditions have the most impact on flight delays?

The weather conditions that have the most impact on flight delays are (in order): Rain (6.90%), Storm (4.20%), and Fog (2.21%).

Business recommendations:

Based on the conclusions above, our business recommendations are to prioritize and focus on days where the weather conditions are either rain, storm, or fog. Rain is the most impactful weather condition on flight delays.

The costs of delays for an airline include direct costs to passengers, lost demand, and indirect costs, and to minimize these costs, we recommend the airline to pay close attention to these 3 (rain, storm, fog) weather conditions, and to implement an **inflated schedule time** for flights where any of these weather conditions are present.