

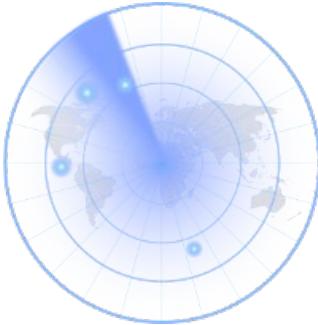
# Retrieving and Visualizing Real-time data from API



Ana Maria Cuciuc  
Polina Voroshylova  
Tarazali Ryskul

# Project Structure

1. The data sets sources and data description
2. Airline Crash Analytics with Python
3. Hadoop and Spark
4. Data preparation and data processing
5. Data Visualization
6. Project reflections



# The Open Sky API

REST API

GET /states/all

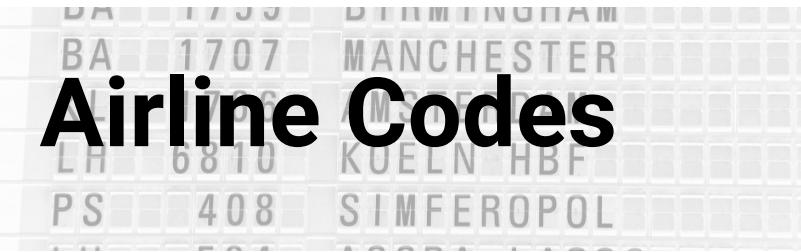
Python API

```
class opensky_api.OpenSkyApi()
```

Java API

```
OpenSkyStates os =  
    api.getMyStates(0, null,  
    null);
```

# Airline Codes



source: wikipedia.com

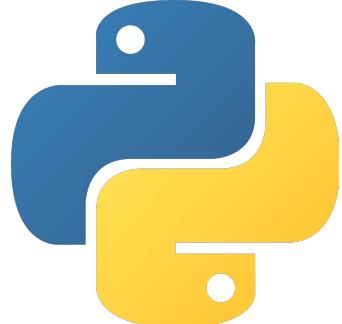
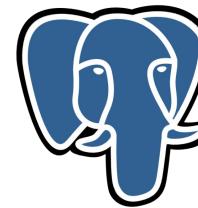
# Airline Crashes



source: socrata.com

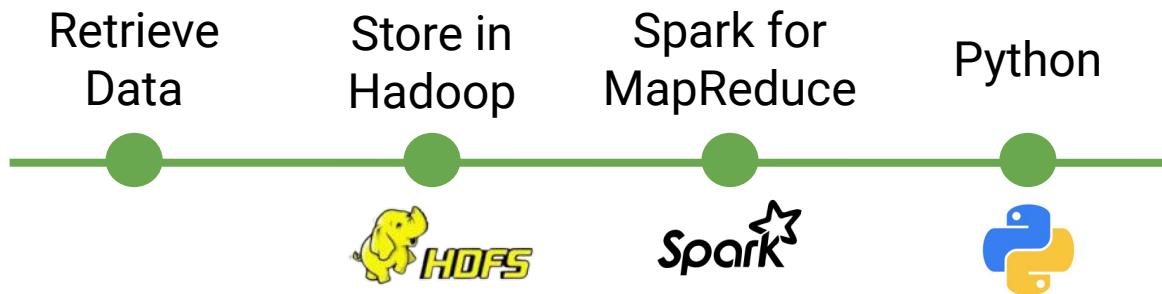
- + Contains very detailed data
- Data is old: last entry in 2009
- A lot of missing values

# Technologies



# How we proceeded?

Crashes Data



# Storing in Hadoop

```
tarazali_ryskul@seneca3:~/airplane$ cd
tarazali_ryskul@seneca3:~$ hadoop fs -put airplane/Airplane_Crashes_Since_1908.csv airplane_hdfs/
tarazali_ryskul@seneca3:~$ hadoop fs -ls airplane_hdfs
Found 1 items
rw-r--r--  3 tarazali_ryskul student      1600720 2019-06-22 15:35 airplane_hdfs/Airplane_Crashes_Si
ce 1908.csv
```

## Browse Directory

/user/tarazali_ryskul/airplane_hdfs								Go!								
Show		25	entries					Search:								
<input type="checkbox"/>		Permission		Owner		Group		Size		Last Modified		Replication		Block Size		Name
<input type="checkbox"/>	-rw-r--r--		tarazali_ryskul	student			1.53 MB		Jun 22 15:35		3		128 MB		Airplane_Crashes_Since_1908.csv	
Showing 1 to 1 of 1 entries														Previous	1	Next

# Spark

```
File "/opt/cloudera/parcels/CDH-6.2.0-1.cdh6.2.0.p0.967373/lib/spark/python/pyspark/sql/utils.py",
line 69, in deco
    raise AnalysisException(s.split(': ', 1)[1], stackTrace)
pyspark.sql.utils.AnalysisException: u'Path does not exist: hdfs://seneca1.lehre.hwr-berlin.de:8020/
user/tarazali_rvskul/Airplane_Crashes_Since_1908.csv:'
>>> airplane = spark.read.csv("airplane_hdfs/Airplane_Crashes_Since_1908.csv", header=True, inferSchema=True)
>>> airplane.count()
5268
>>> airplane.schema()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'StructType' object is not callable
>>> airplane.printSchema()
root
 |-- Date: string (nullable = true)
 |-- Time: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Operator: string (nullable = true)
 |-- Flight #: string (nullable = true)
 |-- Route: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Registration: string (nullable = true)
 |-- cn/In: string (nullable = true)
 |-- Aboard: integer (nullable = true)
 |-- Fatalities: integer (nullable = true)
 |-- Ground: integer (nullable = true)
 |-- Summary: string (nullable = true)

>>> |
```

# Location-Operator

Location	Operator	avg(Fatalities)
Chicago, Illinois	Chicago-Detroit A...	1.0
Sir Lowry's pass, ...	Union Airways	3.0
Haydens Peak, Wyo...	United Air Lines	19.0
Bucharest, Romania	LARES	15.0
English Channel, ...	British Overseas ...	9.0
Castel Benito, Libya	Skyways of London	1.0
Gao, French West ...	Société Aérienne ...	17.0
Near San Felipe, ...	TACA	12.0
Christchurch, New...	Straits Air Freig...	4.0
Near Manila, Phil...	Philippine Air Lines	1.0
East Nassau, New ...	Chester Airport	3.0
Near Shanghai, China	CAAC	40.0
Guangzhou, China	Total	8.0
Java Sea, Indonesia	Pelita Air Service	9.0
Near Tamworth, A...	Jetcraft	1.0
Sao Gabriel, Brazil	Rico Taxi Aero	8.0
Matsu Island, Taiwan	Formosa Airlines	16.0
AtlantiOcean, off...	Renan Airways	5.0
Bluefields, Nicar...	Fuerza Aérea Nica...	28.0
Near Puerto Cabel...	Private - OverteC...	13.0

# Location Splitting

```
|          Location|  
+-----+  
|Schievelbein, Ger...|  
|Near Walsenberg, ...|  
| Schiphol, Amsterdam|  
| Sweetwater, Texas|  
|Mt. Banahaur, Phi...|  
|Braemar Reservoir,...|  
|  Caracas, Venezuela|  
|Williamsport, Penn...|  
|Near San Francisc...|  
|off Tangiers, Mor...|  
|Near Hasloh, Germany|  
|Sierra de Atalaya...|  
| Straits of Johore...|  
|    Nightmute, Alaska|  
|Near Cuenca, Azua...|  
|Wilmington, North...|  
|950 nm S of Shem...|  
|     Arequipa, Peru|  
|Yangon (Rangoon),...|  
|Off Mannar, Sri L...|  
+-----+  
only showing top 20 rows
```

```
AttributeError: module 'object' has no attribute 'functions'  
>>> import pyspark.sql.functions  
>>> split_col = pyspark.sql.functions.split(airplane['Location'], ',')  
>>> airplane = airplane.withColumn('Country', split_col.getItem(1))  
>>> airplane.show()  
+-----+-----+-----+-----+-----+-----+-----+  
|  Date| Time|      Location|       Operator|Flight #|      Route|  
|Type|Registration|cn/In|Aboard|Fatalities|Ground|Summary|Country|  
+-----+-----+-----+-----+-----+-----+-----+  
|09/17/1908|17:18| Fort Myer, Virginia|Military - U.S. Army| null|Demonstration| Wright Flyer  
|III| null| 1| 2| 1| 0|During a demonstr...| null| Virginia|  
|07/12/1912|06:30|Atlantic City, New ...|Military - U.S. Navy| null| Test flight| Dirigible|  
| null| null| 5| 5| 0|First U.S. dirigi...| null| New Jersey|  
|08/06/1913| null|Victoria, British...| Private| -| null| Curtiss seaplane|  
| null| null| 1| 1| 0|The first fatal a...| British Columbia|  
|09/09/1913|18:30| Over the North Sea|Military - German...| null| null| Zeppelin L-1 (air...|  
| ...| null| null| 20| 14| 0|The airship flew ...| null| null|  
|10/17/1913|10:30|Near Johannisthal,...|Military - German...| null| null| Zeppelin L-2 (air...|  
| ...| null| null| 30| 30| 0|Hydrogen gas whic...| null| Germany|  
|03/05/1915|01:00| Tienen, Belgium|Military - German...| null| null| Zeppelin L-8 (air...|
```

# Country RDD

```
>>> country = airplane.select('Country').collect()
>>> country.show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'list' object has no attribute 'show'
>>> country
[Row(Country=u' Virginia'), Row(Country=u' New Jersey'), Row(Country=None), Row(Country=u' Germany'), Row(Country=u' Belgium'), Row(Country=u' Bulgeria'), Row(Country=u' England'), Row(Country=u' England'), Row(Country=u' England'), Row(Country=u' Belgium'), Row(Country=None), Row(Country=u' North Sea'), Row(Country=None), Row(Country=u' France'), Row(Country=u' Germany'), Row(Country=u' North Sea'), Row(Country=u' New Jersey')]
```

```
>>> df_count = spark.createDataFrame(country).show()
+-----+
|Country|
+-----+
| Virginia|
| New Jersey|
| British Columbia|
| null|
| Germany|
| Belgium|
| Germany|
| Bulgeria|
| England|
| England|
| Germany|
| England|
| Belgium|
| null|
| North Sea|
| North Sea|
| null|
| France|
| null|
| Germany|
+-----+
only showing top 20 rows
```

# Country MapReduce

```
>>> text_file = sc.textFile("airplane_hdfs/country.txt")
>>> text_file.collect()
[u'Virginia', u'New Jersey', u'British Columbia', u'....', u'Germany',
Igium', u'....', u'North Sea', u'North Sea', u'....', u'France', u'....',
, u'....', u'New Jersey', u'Indiana', u'New Jersey', u'New Jersey',
ania', u'France', u'Spain', u'Illinois', u'Wyoming', u'England',
ng', u'California', u'France', u'England', u'France', u'Austral
Italy', u'Morocco', u'Morocco', u'France', u'Spain', u'Wyoming',
Pennsylvania' '....' u'Colombia' '....' u'France' '....' u'Indiana'
```

```
>>> wor = text_file.flatMap(lambda line:line.split("\n"))
>>> wor.collect()
[u'Virginia', u'New Jersey', u'British Columbia', u'....', u'Germany',
Igium', u'....', u'North Sea', u'North Sea', u'....', u'France', u'....',
, u'....', u'New Jersey', u'Indiana', u'New Jersey', u'New Jersey', u'',
ania', u'France', u'Spain', u'Illinois', u'Wyoming', u'England', u'',
ng', u'California', u'France', u'England', u'France', u'Austral
Italy', u'Morocco', u'Morocco', u'France', u'Spain', u'Wyoming',
Pennsylvania', u'....', u'Colombia', u'....', u'France', u'....', u'Indiana',
, u'France', u'Ohio', u'France', u'Ohio', u'Germany', u'....', u'Czech
u' Oregon', u'France', u'Morocco', u'Qld', u'Idaho', u'Indiana', u'',
Germany', u'Mauritania', u'New Jersey', u'France', u'....', u'Indiana',
, u'Brazil', u'France', u'Wyoming', u'Spain', u'....', u'Minnesota', u'
```

```
>>> word_tup = wor.map(lambda word: (word,1))
>>> word_tup.collect()
[(u'Virginia', 1), (u'New Jersey', 1), (u'British
    1), (u'England', 1), (u'Germany', 1), (u'England'
    1), (u'Germany', 1), (u'North Sea', 1), (u'New Je
    , 1), (u'Indiana', 1), (u'New Jersey', 1), (u'New
    y', 1), (u'Ohio', 1), (u'Pennsylvania', 1), (u'Fra
    ), (u'Wisconsin', 1), (u'....', 1), (u'Nevada', 1),
    1), (u'France', 1), (u'Australia', 1), (u'Franc
```

```
>>> counts_word = word_counts.map(lambda (x,y): (y,x))
>>> counts_word.collect()
[(47, u'Canada'), (1, u'East Timor'), (12, u'Mississippi'),
, u'Lithuania'), (4, u'Cambodia'), (19, u'Switzerland'), (1,
, (19, u'Louisiana'), (1, u'Republioof Djibouti'), (51, u'Tex
(8, u'Azores'), (1, u'Channel Islands'), (1, u'Da Nang'), (1,
u'Germany'), (1, u'UAE'), (14, u'Panama'), (1, u'Dominica')
u'Tanzania'), (1, u'Rhodesia (Zimbabwe)'), (1, u'Zulia')]
```

# Country MapReduce

```
>>> word_counts = word_tup.reduceByKey(lambda x,y:x+y)
>>> word_counts.collect()
[(u'Canada', 47), (u'near Point Barrow', 1), (u'Mississippi', 12), (u'Antigua', 1), (u'WA', 2), (u'Virginia.', 1), (u'East Germany', 4), (u'Hrvatska', 1), (u'Lithuania', 1), (u'Cambodia', 4), (u'110 miles West of Ireland', 1), (u'Covington', 1), (u'Bosnia Herzegovina', 1), (u'Kent', 2), (u'Argentina', 44), (u'Louisiana', 19), (u'India / Kandahar', 1), (u'Texas', 51), (u'Ghana', 2), (u'Romania', 1), (u'Saudi Arabia', 15), (u'near Sibiu', 1), (u'Bugaria', 1), (u'Azores', 8), (u'N of Santander', 1), (u'Channel Islands', 1), (u'Orly', 1), (u'Cailifornia', 1), (u'U.S. Virgin Islands', 2), (u'Guatemala', 17), (u'Wisconsin', 3), (u'Germany', 78), (u'Timor', 2), (u'Panama', 14), (u'Spain', 60), (u'Western Samoa', 1), (u'Island of Madeira', 1), (u'Miami', 1), (u'Tanzania', 6), (u'Republica (Zimbabwe)', 1), (u'Gabon', 5), (u'Luxembourg', 2), (u'West Vlaanderen', 1), (u'Thiells', 1), (u'Samoa', 1), (u'New Zealand', 24), (u'Dominican Republic', 5), (u'Jersey', 2), (u'England', 85), (u'Brisbane', 1), (u'Belgium Congo', 1), (u'Guam', 4), (u'India', 95), (u'off Ustica', 1), (u'Hati', 1), (u'New Jersey', 40), (u'Africa', 1), (u'London', 3), (u'Arkansas', 13), (u'South Korea', 17), (u'Tajikistan', 1), (u'Great Inagua', 1), (u'Turkey', 31), (u'British Virgin Islands', 1), (u'Caribbean', 1), (u'Ireland', 14), (u'Lombardia', 1), (u'Solomon Islands', 2), (u'Washington', 43), (u'Saint Lucia', 1), (u'West Germany', 8), (u'Kyrgyzstan', 2), (u'Mongolia', 6), (u'18 NNW of Benton Harbor', 1), (u'Inner Mongolia', 1), (u'Lousiana', 1), (u'Slovakia', 2), (u'Peru', 52), (u'Maryland', 21), (u'Norway', 29), (u'Domincan Republic', 1), (u'Cook Islands', 1), (u'NWT', 1), (u'Uzbekistan', 9), (u'Jiangsu', 1), (u'Wisconsin', 9), (u'Sumarta', 1), (u'Lombok Island', 1), (u'Huanuco', 1), (u'Cameroon', 12), (u'Michigan', 27), (u'Oregon', 16), (u'near Ft. Collins', 1), (u'Coloado', 1), (u'China', 78), (u'WV', 1), (u'Los Angeles', 2), (u'Bahrain', 1), (u'West Indies', 1), (u'Tasmania', 2), (u'British Columbia Canada', 1), (u'Djibouti', 1), (u'French Polynesia', 2)

>>> sorted.take(50)
[(208, u''), (178, u'Brazil'), (173, u'Alaska'), (169, u'Russia'), (145, u'Colombia'), (138, u'California'), (122, u'France'), (95, u'India'), (85, u'England'), (82, u'Indonesia'), (80, u'Mexico'), (78, u'Germany'), (78, u'China'), (73, u'Italy'), (68, u'Philippines'), (60, u'Spain'), (59, u'New York'), (56, u'Venezuela'), (52, u'Australia'), (52, u'Peru'), (51, u'Texas'), (50, u'Ohio'), (47, u'Canada'), (46, u'Bolivia'), (44, u'Argentina'), (43, u'Washington'), (42, u'Pennsylvania'), (41, u'Japan'), (41, u'Florida'), (41, u'Angola'), (40, u'New Jersey'), (39, u'Illinois'), (39, u'Colorado'), (38, u'Egypt'), (37, u'South Vietnam'), (36, u'Georgia'), (35, u'Iran'), (35, u'Taiwan'), (34, u'Vietnam'), (33, u'Ecuador'), (32, u'Hawaii'), (31, u'Arizona'), (31, u'USSR'), (31, u'Turkey'), (31, u'Nigeria'), (30, u'Sudan'), (29, u'Pakistan'), (29, u'Afghanistan'), (29, u'Norway'), (28, u'Nevada')]
```

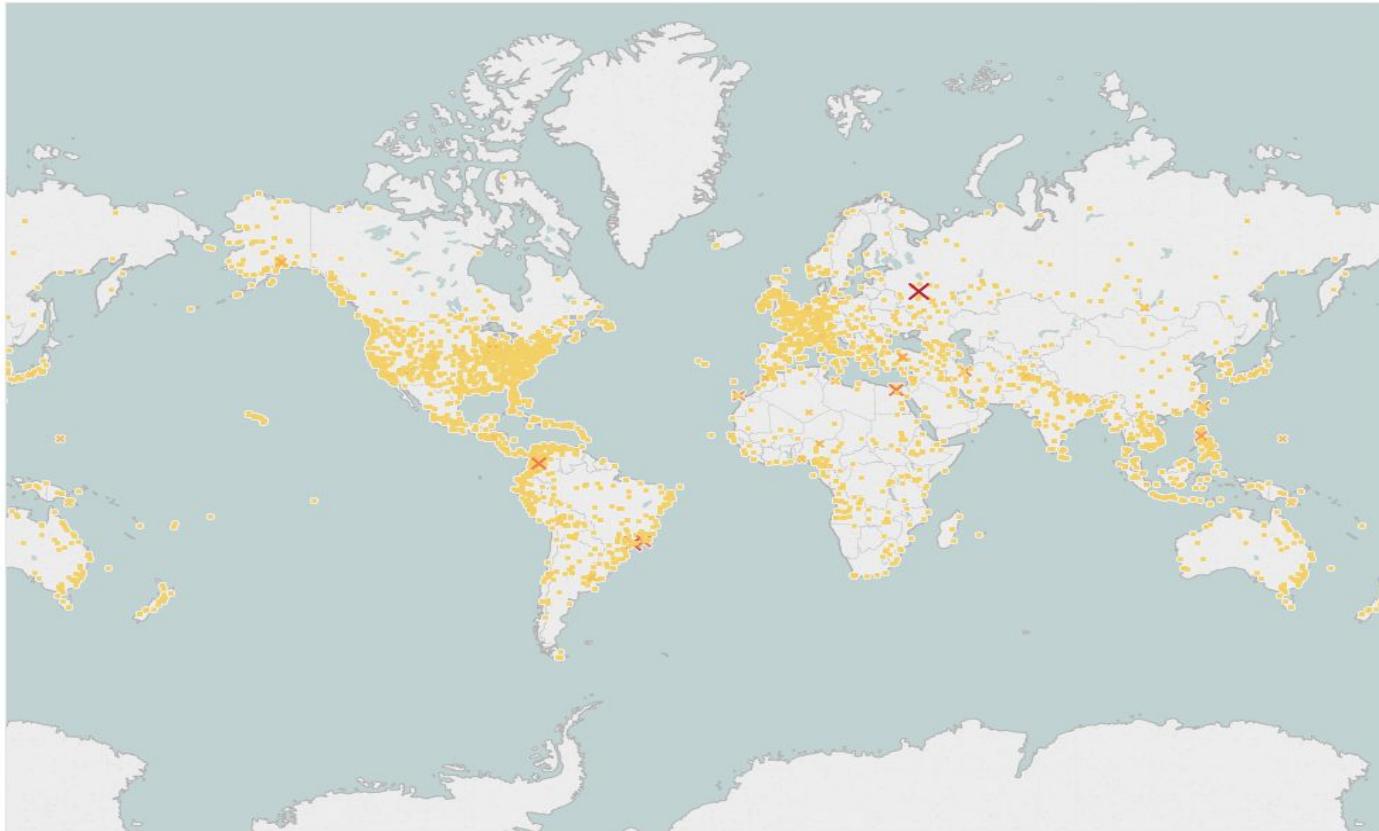
# Distributed Storage

## Browse Directory

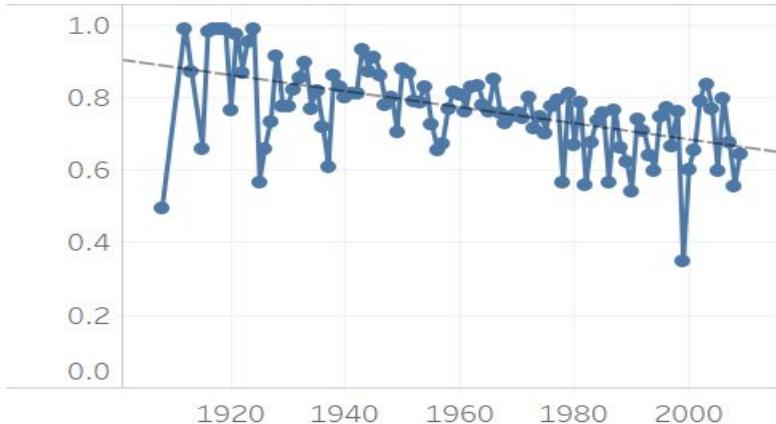
/user/tarazali_ryskul/Country								Go!				
Show	25	entries							Search:			
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name				
<input type="checkbox"/>	-rw-r--r--	tarazali_ryskul	student	0 B	Jun 22 20:37	3	128 MB	_SUCCESS				
<input type="checkbox"/>	-rw-r--r--	tarazali_ryskul	student	43.79 KB	Jun 22 21:05	3	128 MB	country.txt				
<input type="checkbox"/>	-rw-r--r--	tarazali_ryskul	student	25.12 KB	Jun 22 20:37	3	128 MB	part-00000-1506decb-421e-42f2-8de6-9fc8f4cba87e-c000.csv				
<input type="checkbox"/>	-rw-r--r--	tarazali_ryskul	student	18.67 KB	Jun 22 20:37	3	128 MB	part-00001-1506decb-421e-42f2-8de6-9fc8f4cba87e-c000.csv				
Showing 1 to 4 of 4 entries								Previous	1	Next		

# Data Visualization

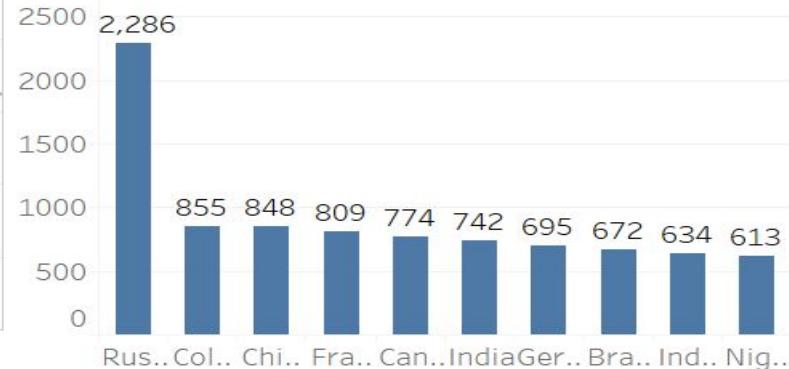
Spatial distribution of airplane crashes (1908-2009)



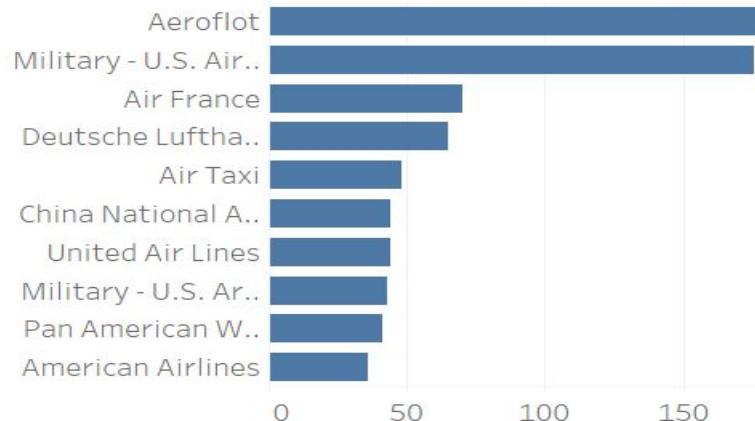
### The percentage of fatality



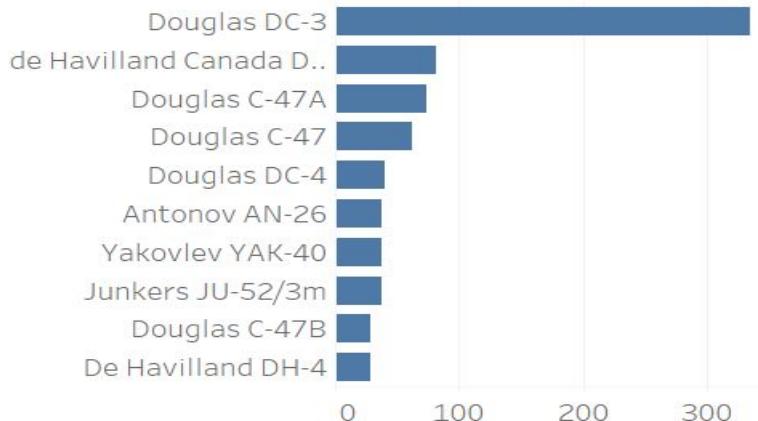
### Top countries with highest fatalities



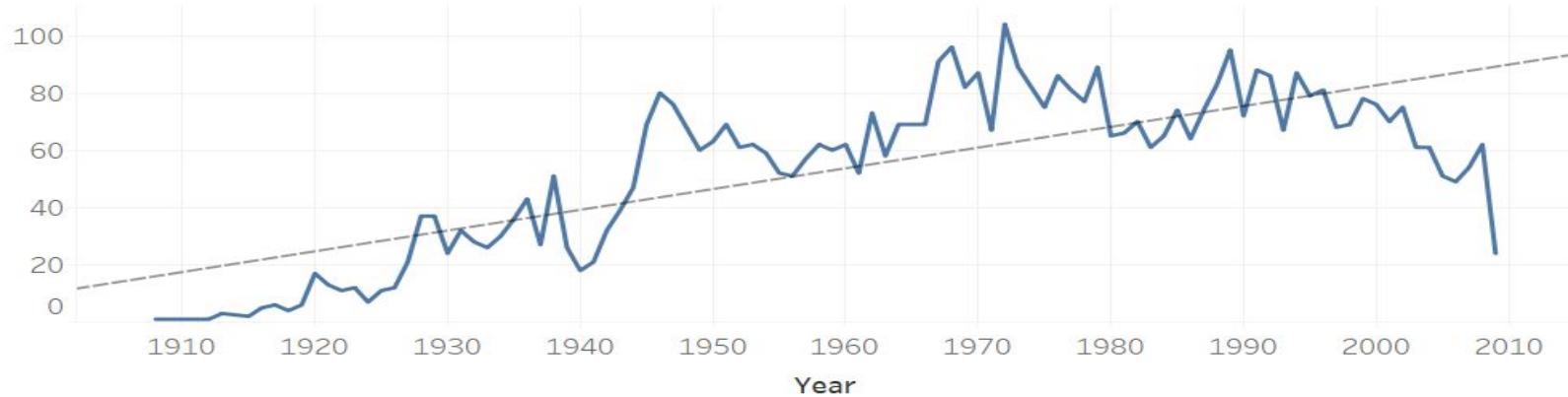
### Airlines involved-Top 10



### Aircraft Type-Top 10



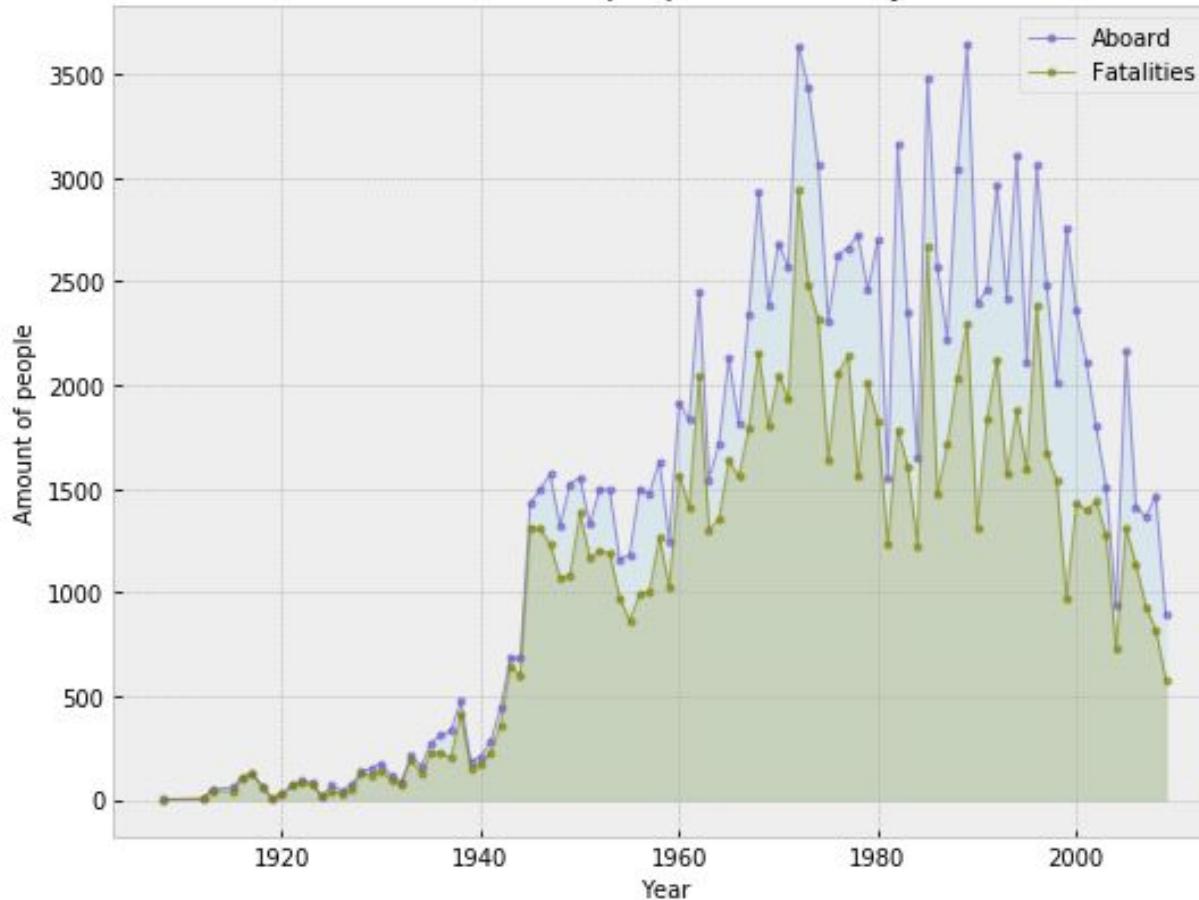
### Total number of crashes per Year



### Total number of crashes per Month



### Total number of people involved by Year



# Topic Modelling

## Preprocessing

```
[['demonstration', 'flight', 'army', 'flyer', 'fly'  
'kill', 'lt', 'thomas', 'selfridge', 'passenger',  
'ight', 'tear', 'loose', 'wire', 'brace', 'rudder',  
'ribs', 'pelvi', 'leg', 'selfridge', 'suffer', 'cru
```

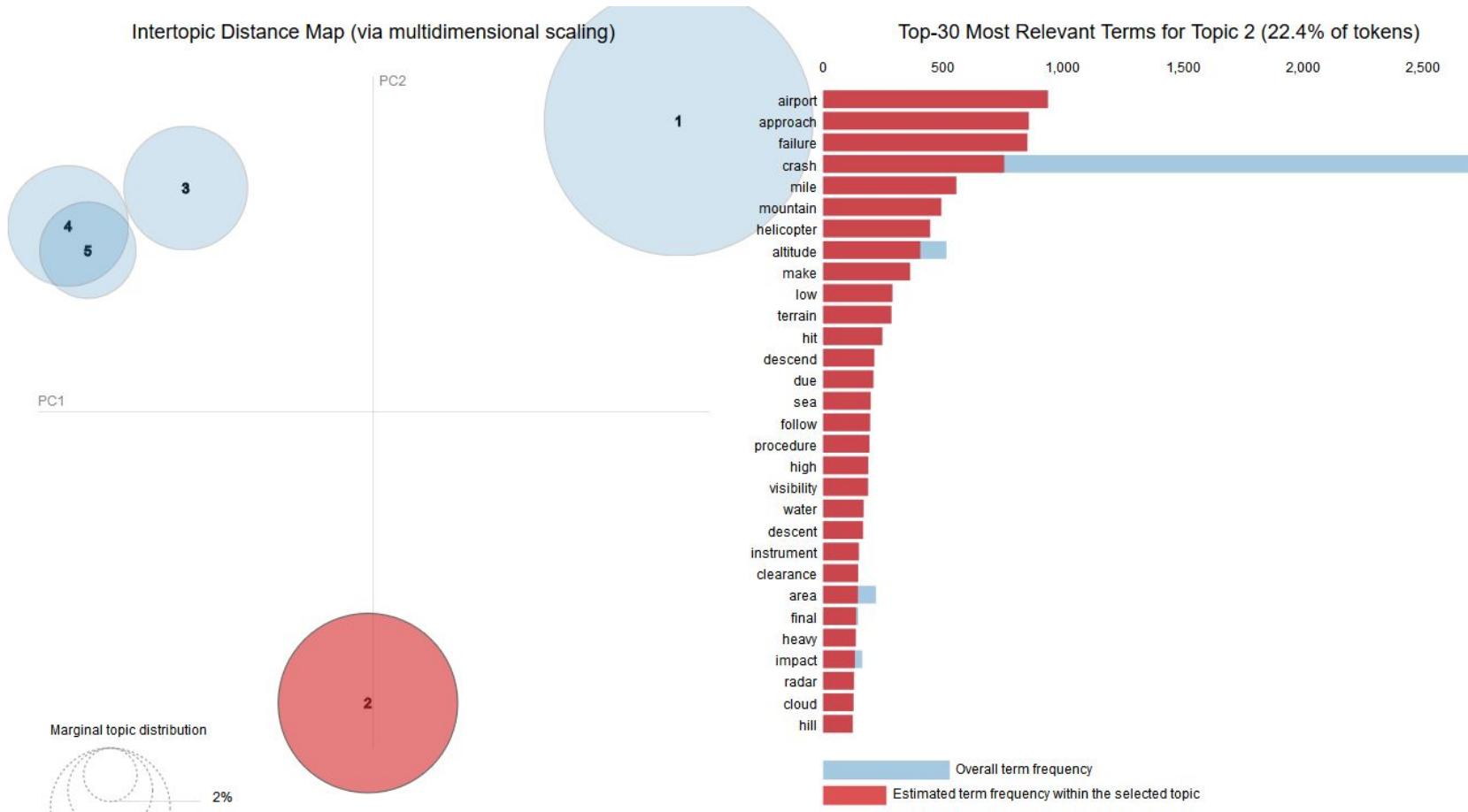
## Corpus Dictionary

```
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1),  
1), (16, 1), (17, 1), (18, 1), (19, 1), (2  
(30, 1), (31, 1), (32, 1), (33, 1), (34, 2
```

Pilot faults and  
engine issues

```
[(0,  
'0.059*"plane" + 0.047*"crash" + 0.039*"pilot" + 0.025*"runway" + '  
'0.023*"take" + 0.021*"engine" + 0.020*"crew" + 0.019*"land" + '  
'0.016*"attempt" + 0.014*"flight")),  
(1,  
'0.067*"flight" + 0.056*"condition" + 0.055*"weather" + 0.043*"takeoff" + '  
'0.036*"poor" + 0.031*"fog" + 0.019*"ice" + 0.015*"side" + 0.013*"burst" + '  
'0.012*"destroy")),
```

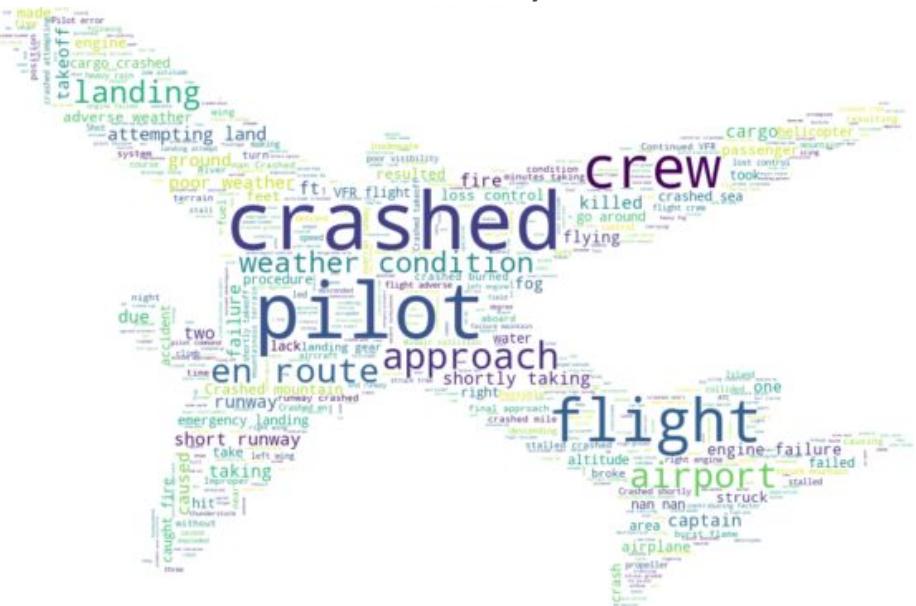
Weather  
conditions



## Location of Accident



## Brief Summary



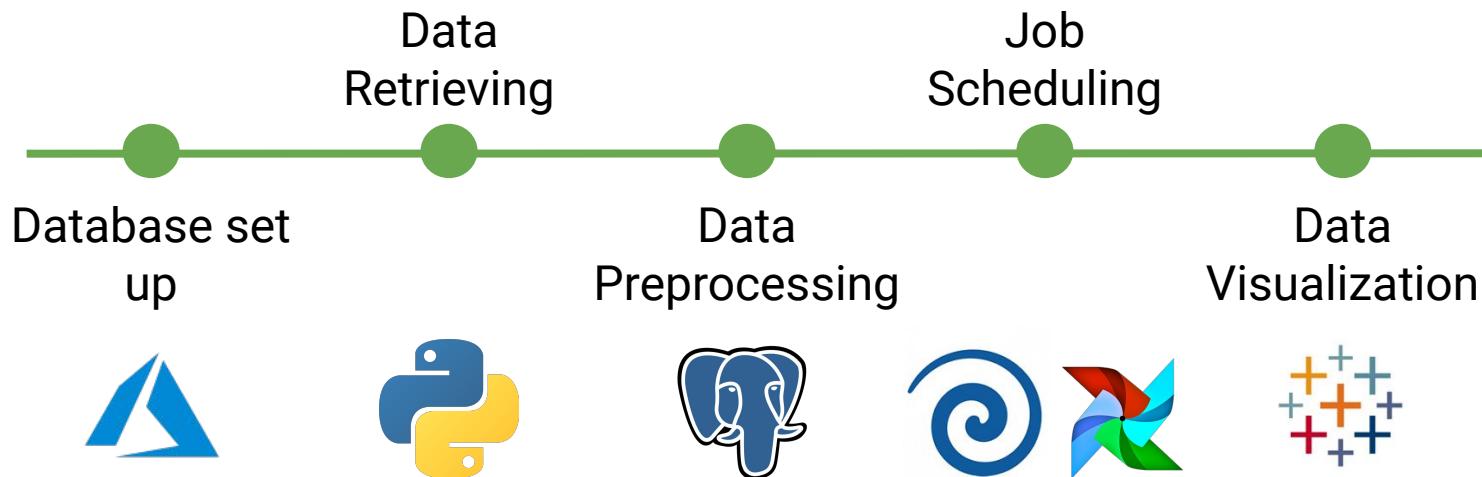


# Data Cleaning

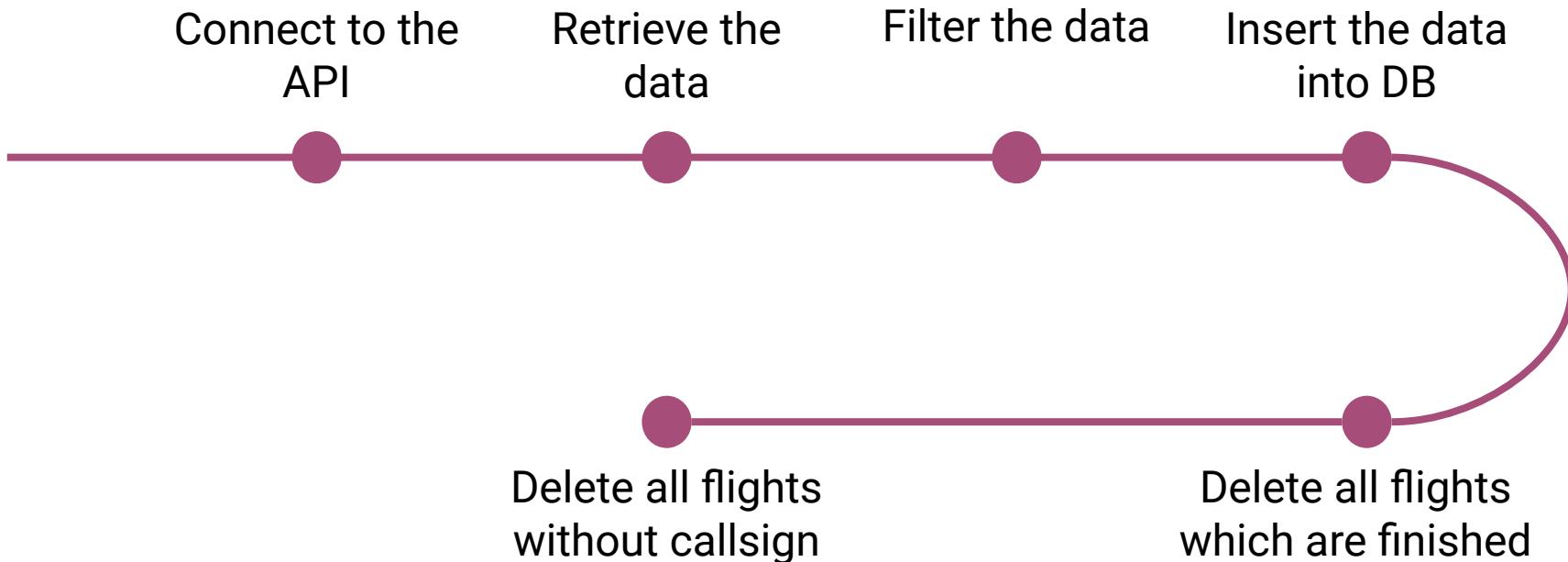
1. Missing Values
2. Splitting date column in month and year
3. Splitting the location column in city and country

# How we proceeded?

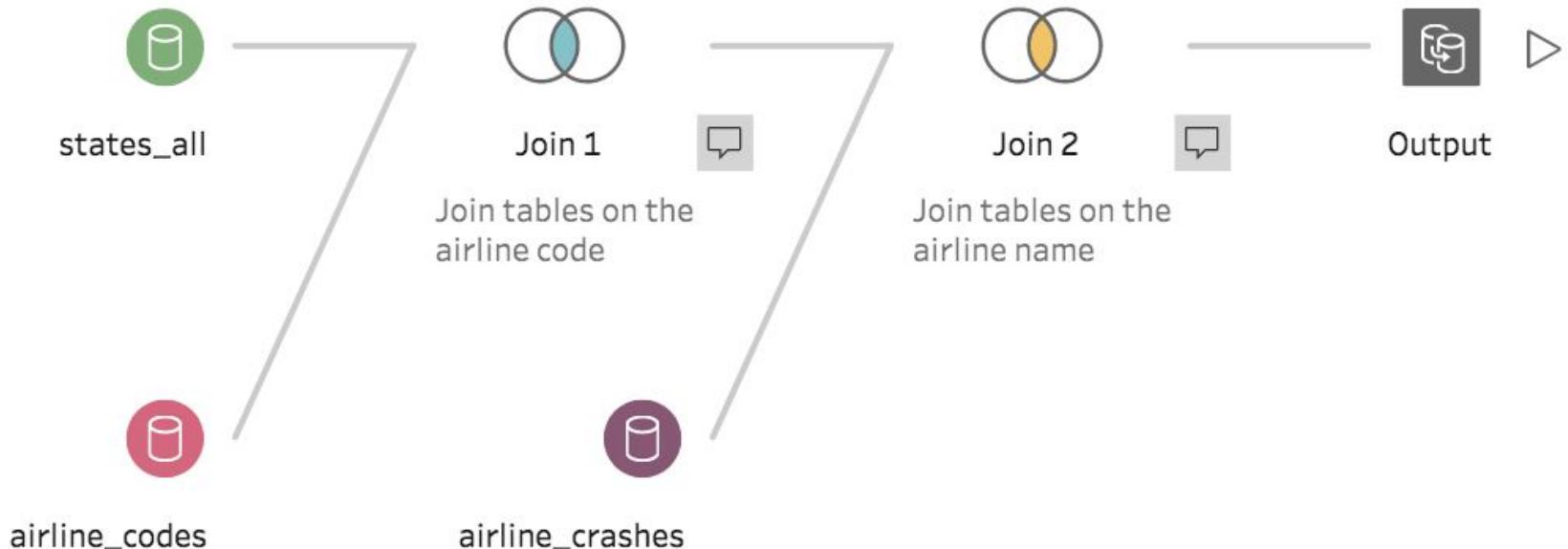
Flights Data



# Job scheduler flow



# Data Preprocessing

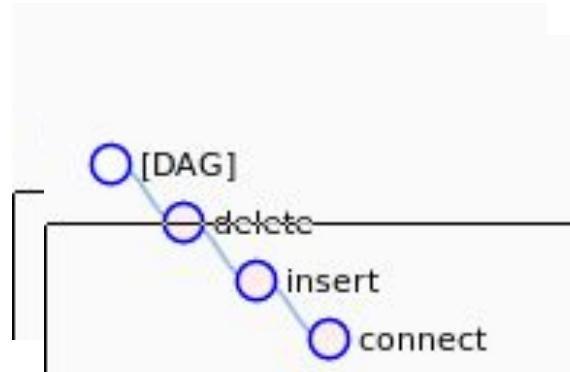


Airflow

DAGs Data Profiling ▾

## DAGs

		DAG	Schedule
		opensky_retrieval_api	0 12 * * *



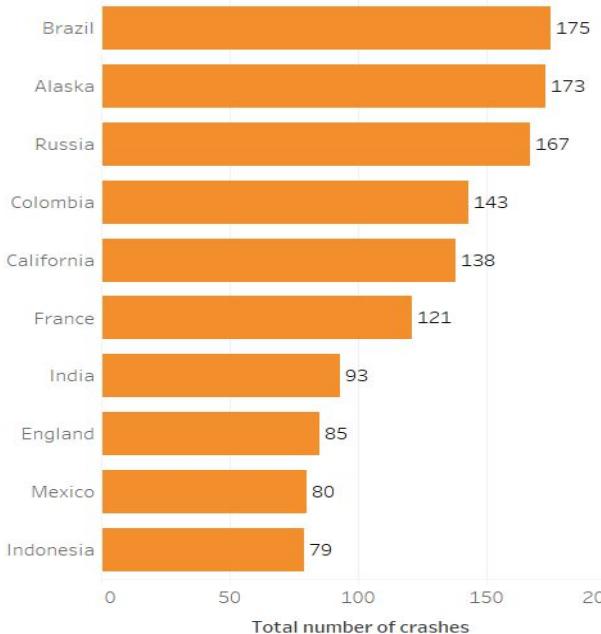
# states table



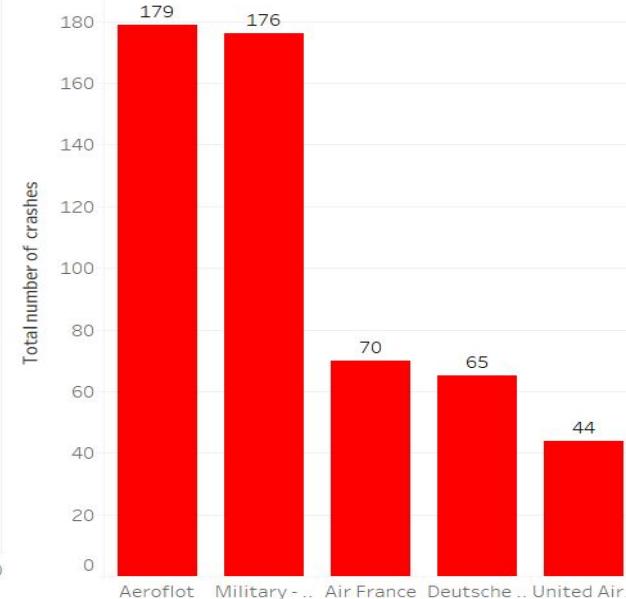
# airline\_crashes table

Exploratory analysis for Airline Crashes table

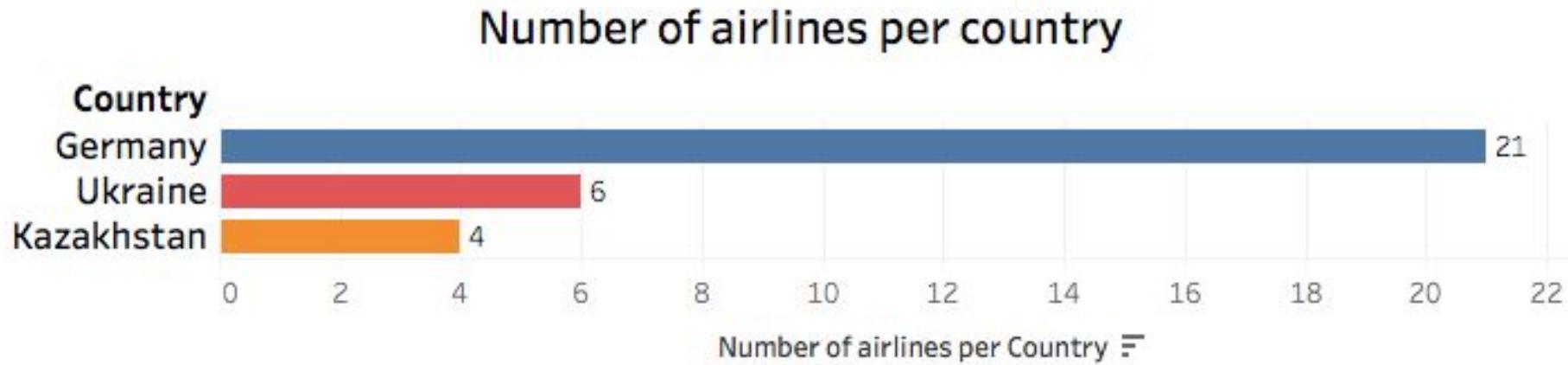
Top countries per number of crashes



Top 5 airlines per number of crashes



# airline\_codes table



# Job Scheduling



Off DAG: opensky\_retrieval\_api Retrieving data from API

Graph View Tree View Task Duration Task Tries La

Refresh Delete

None Base date: 2019-07-11 18:10:15 Number of runs: 25 Run:

PythonOperator

connect → insert → delete

A screenshot of the Airflow web interface showing a DAG named "opensky\_retrieval\_api". The DAG is currently off. It has four tasks: "Graph View", "Tree View", "Task Duration", and "Task Tries". Below these are buttons for "Refresh" and "Delete". A base date of "2019-07-11 18:10:15" and a number of runs set to 25 are displayed. A "Run:" button is present. A single task, "PythonOperator", is listed. At the bottom, a sequence of three rounded rectangular boxes contains the text "connect", "insert", and "delete" separated by arrows.

Job Scheduling

Repeat:

Type: Interval

Interval in seconds: 0

Interval in minutes: 20

Time of day: 12 0

Day of week: Monday

Day of month: 1

Help OK Cancel

A screenshot of a "Job Scheduling" dialog box. It includes fields for "Repeat" (checked), "Type" (set to "Interval"), "Interval in seconds" (0), "Interval in minutes" (20), "Time of day" (12:00), "Day of week" (Monday), and "Day of month" (1). At the bottom are "Help", "OK", and "Cancel" buttons.

# Data Visualization



# Project Reflexions

1. Limited Capacity
2. Old flights crashes data
3. Not enough data about airports and destinations

# Q&A

 [https://github.com/vppolina/flights\\_project](https://github.com/vppolina/flights_project)

