Working Steps:

1. download data in any format

2. creating schema

3. data processing using pyspark

4. visualize the output

Reference:

[**https://github.com/Amitdeb008/Big-Data-Projects/tree/master/Airline%20Analysis**](https://github.com/Amitdeb008/Big-Data-Projects/tree/master/Airline%20Analysis)

[**https://github.com/lpatruno/airline-time-analysis**](https://github.com/lpatruno/airline-time-analysis)

[**https://github.com/SaritaIngu/US-FlightDelayAnalysis**](https://github.com/SaritaIngu/US-FlightDelayAnalysis)

**https://github.com/IBM/acme-freight-erp**

<https://github.com/ksatola/US-Airline-Delay-Statistics/blob/master/FlightPaths2008.ipynb> **-for data wangling**

US Domestic Flight Cancellations Dataset:

<https://www.kaggle.com/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018/data>

Download size: 7 GB

**Schema for spark:**

schema= StructType(

[

StructField(“FL\_DATE”,DateType(), Fasle),

StructField(“OP\_CARRIER”,StringType(), Fasle),

StructField(“OP\_CARRIER\_FL\_NUM”,IntegerType(), Fasle),

StructField(“ORIGIN”,StringType(), Fasle),

StructField(“DEST”,StringType(), Fasle),

StructField(“CRS\_DEP\_TIME”,IntegerType(), Fasle),

StructField(“DEP\_TIME”,DoubleType(), Fasle),

StructField(“DEP\_DELAY”,DecimalType(), Fasle),

StructField(“TAXI\_OUT”,DoubleType(), Fasle),

StructField(“WHEELS\_OFF”,DoubleType(), Fasle),

StructField(“WHEELS\_ON”,DoubleType(), Fasle),

StructField(“TAXI\_IN”,DoubleType(), Fasle),

StructField(“CRS\_ARR\_TIME”,IntegerType(), Fasle),

StructField(“DEP\_TIME”,DoubleType(), Fasle),

StructField(“ARR\_TIME”,DoubleType(), Fasle),

StructField(“ARR\_DELAY”,DecimalType(), Fasle),

StructField(“CANCELLED”,DecimalType(), Fasle),

StructField(“CANCELATION\_CODE”,STRINGType(), True),

StructField(“DIVERTED”,DecimalType(), Fasle),

StructField(“CSR\_ELAPSED\_TIME”,DecimalType(), Fasle),

StructField(“ACTUAL\_ELAPSED\_TIME”,DecimalType(), Fasle)

]

)

data.printSchema()

root

|-- fl\_date: string (nullable = true)

|-- op\_carrier: string (nullable = true)

|-- op\_carrier\_fl\_num: string (nullable = true)

|-- origin: string (nullable = true)

|-- dest: string (nullable = true)

|-- crs\_dep\_time: string (nullable = true)

|-- dep\_time: string (nullable = true)

|-- dep\_delay: string (nullable = true)

|-- taxi\_out: string (nullable = true)

|-- wheels\_off: string (nullable = true)

|-- wheels\_on: string (nullable = true)

|-- taxi\_in: string (nullable = true)

|-- crs\_arr\_time: string (nullable = true)

|-- arr\_time: string (nullable = true)

|-- arr\_delay: string (nullable = true)

|-- cancelled: string (nullable = true)

|-- cancellation\_code: string (nullable = true)

|-- diverted: string (nullable = true)

|-- crs\_elapsed\_time: string (nullable = true)

|-- actual\_elapsed\_time: string (nullable = true)

|-- air\_time: string (nullable = true)

|-- distance: string (nullable = true)

|-- carrier\_delay: string (nullable = true)

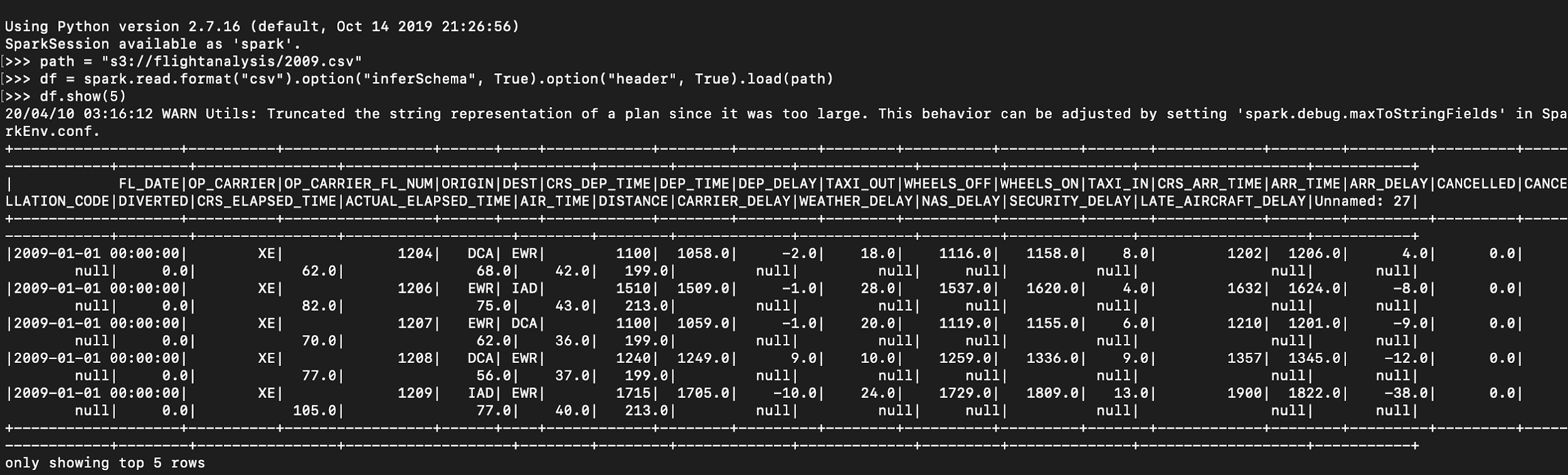
|-- weather\_delay: string (nullable = true)

|-- nas\_delay: string (nullable = true)

|-- security\_delay: string (nullable = true)

|-- late\_aircraft\_delay: string (nullable = true)

**Using EMR and S3 bucket**



Need: to load all csv files into a single dataframe

<https://towardsdatascience.com/getting-started-with-pyspark-on-amazon-emr-c85154b6b921>

Once the cluster is turned on, start notebook → this will be saved in S3 buckets and we can continue to work with the same notebook powered by different clusters.

Data Cleaning:

After removing unnamed columns, we have 19 columns worth of data (original 28)

In case notebook cannot load pandas:

<https://aws.amazon.com/blogs/big-data/install-python-libraries-on-a-running-cluster-with-emr-notebooks/>