Assignment 3

April 25, 2021

Yuming Chen

320180939611

0.1 Preparation

[1]:
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('assignment3').getOrCreate()
```

[2]:
```
flights_df = spark.read.csv("flights.csv",header=True, inferSchema=True)
weather_df = spark.read.csv("weather-samples.csv",header=True, inferSchema=True)
```

Question 1: Data Exploration in Spark

1. Create a Spark DataFrom from the flights.csv file (provided).
2. Write Spark code to answer the following questions (note: only one code is left to you to fill for each question, but it could have several code lines).

a. List all the unique destination countries. (5%)

[3]:
```
flights_df.select("DEST_COUNTRY_NAME").distinct().collect()
```

[3]:
```
[Row(DEST_COUNTRY_NAME='Chad'),
 Row(DEST_COUNTRY_NAME='Anguilla'),
 Row(DEST_COUNTRY_NAME='Russia'),
 Row(DEST_COUNTRY_NAME='Paraguay'),
 Row(DEST_COUNTRY_NAME='Senegal'),
 Row(DEST_COUNTRY_NAME='Sweden'),
 Row(DEST_COUNTRY_NAME='Kiribati'),
 Row(DEST_COUNTRY_NAME='Guyana'),
 Row(DEST_COUNTRY_NAME='Philippines'),
 Row(DEST_COUNTRY_NAME='Malaysia'),
 Row(DEST_COUNTRY_NAME='Fiji'),
 Row(DEST_COUNTRY_NAME='Turkey'),
 Row(DEST_COUNTRY_NAME='Malawi'),
 Row(DEST_COUNTRY_NAME='Germany'),
 Row(DEST_COUNTRY_NAME='Jordan'),
```

```
Row(DEST_COUNTRY_NAME='Palau'),
Row(DEST_COUNTRY_NAME='Turks and Caicos Islands'),
Row(DEST_COUNTRY_NAME='France'),
Row(DEST_COUNTRY_NAME='Greece'),
Row(DEST_COUNTRY_NAME='Taiwan'),
Row(DEST_COUNTRY_NAME='British Virgin Islands'),
Row(DEST_COUNTRY_NAME='Dominica'),
Row(DEST_COUNTRY_NAME='Algeria'),
Row(DEST_COUNTRY_NAME='Slovakia'),
Row(DEST_COUNTRY_NAME='Macau'),
Row(DEST_COUNTRY_NAME='Argentina'),
Row(DEST_COUNTRY_NAME='Belgium'),
Row(DEST_COUNTRY_NAME='Angola'),
Row(DEST_COUNTRY_NAME='Ecuador'),
Row(DEST_COUNTRY_NAME='Qatar'),
Row(DEST_COUNTRY_NAME='Finland'),
Row(DEST_COUNTRY_NAME='Nicaragua'),
Row(DEST_COUNTRY_NAME='Ghana'),
Row(DEST_COUNTRY_NAME='Peru'),
Row(DEST_COUNTRY_NAME='United States'),
Row(DEST_COUNTRY_NAME='India'),
Row(DEST_COUNTRY_NAME='China'),
Row(DEST_COUNTRY_NAME='Curacao'),
Row(DEST_COUNTRY_NAME='Malta'),
Row(DEST_COUNTRY_NAME='Kuwait'),
Row(DEST_COUNTRY_NAME='Marshall Islands'),
Row(DEST_COUNTRY_NAME='Chile'),
Row(DEST_COUNTRY_NAME='Martinique'),
Row(DEST_COUNTRY_NAME='Cayman Islands'),
Row(DEST_COUNTRY_NAME='Bolivia'),
Row(DEST_COUNTRY_NAME='Nigeria'),
Row(DEST_COUNTRY_NAME='Italy'),
Row(DEST_COUNTRY_NAME='Suriname'),
Row(DEST_COUNTRY_NAME='Norway'),
Row(DEST_COUNTRY_NAME='Spain'),
Row(DEST_COUNTRY_NAME='Cuba'),
Row(DEST_COUNTRY_NAME='Mauritania'),
Row(DEST_COUNTRY_NAME='Guadeloupe'),
Row(DEST_COUNTRY_NAME='Denmark'),
Row(DEST_COUNTRY_NAME='Barbados'),
Row(DEST_COUNTRY_NAME='Ireland'),
Row(DEST_COUNTRY_NAME='Morocco'),
Row(DEST_COUNTRY_NAME='Panama'),
Row(DEST_COUNTRY_NAME='Cape Verde'),
Row(DEST_COUNTRY_NAME='Hong Kong'),
Row(DEST_COUNTRY_NAME='Venezuela'),
Row(DEST_COUNTRY_NAME='Ukraine'),
```

```
Row(DEST_COUNTRY_NAME='Iceland'),
Row(DEST_COUNTRY_NAME='Israel'),
Row(DEST_COUNTRY_NAME='Saint Barthelemy'),
Row(DEST_COUNTRY_NAME='Saint Kitts and Nevis'),
Row(DEST_COUNTRY_NAME='French Polynesia'),
Row(DEST_COUNTRY_NAME='South Korea'),
Row(DEST_COUNTRY_NAME='Gibraltar'),
Row(DEST_COUNTRY_NAME='Uruguay'),
Row(DEST_COUNTRY_NAME='Bonaire, Sint Eustatius, and Saba'),
Row(DEST_COUNTRY_NAME='Mexico'),
Row(DEST_COUNTRY_NAME='Aruba'),
Row(DEST_COUNTRY_NAME='Indonesia'),
Row(DEST_COUNTRY_NAME='Saint Vincent and the Grenadines'),
Row(DEST_COUNTRY_NAME='The Bahamas'),
Row(DEST_COUNTRY_NAME='Guatemala'),
Row(DEST_COUNTRY_NAME='Azerbaijan'),
Row(DEST_COUNTRY_NAME='Sint Maarten'),
Row(DEST_COUNTRY_NAME='Grenada'),
Row(DEST_COUNTRY_NAME='Federated States of Micronesia'),
Row(DEST_COUNTRY_NAME='Liberia'),
Row(DEST_COUNTRY_NAME='Tunisia'),
Row(DEST_COUNTRY_NAME='Honduras'),
Row(DEST_COUNTRY_NAME='Trinidad and Tobago'),
Row(DEST_COUNTRY_NAME='Saudi Arabia'),
Row(DEST_COUNTRY_NAME='French Guiana'),
Row(DEST_COUNTRY_NAME='Switzerland'),
Row(DEST_COUNTRY_NAME='Ethiopia'),
Row(DEST_COUNTRY_NAME='Latvia'),
Row(DEST_COUNTRY_NAME='Jamaica'),
Row(DEST_COUNTRY_NAME='United Arab Emirates'),
Row(DEST_COUNTRY_NAME='Saint Lucia'),
Row(DEST_COUNTRY_NAME='Canada'),
Row(DEST_COUNTRY_NAME='Samoa'),
Row(DEST_COUNTRY_NAME='Czech Republic'),
Row(DEST_COUNTRY_NAME='Cook Islands'),
Row(DEST_COUNTRY_NAME='Brazil'),
Row(DEST_COUNTRY_NAME='Belize'),
Row(DEST_COUNTRY_NAME='Antigua and Barbuda'),
Row(DEST_COUNTRY_NAME='Dominican Republic'),
Row(DEST_COUNTRY_NAME='Japan'),
Row(DEST_COUNTRY_NAME='Luxembourg'),
Row(DEST_COUNTRY_NAME='New Zealand'),
Row(DEST_COUNTRY_NAME='Greenland'),
Row(DEST_COUNTRY_NAME='Haiti'),
Row(DEST_COUNTRY_NAME='Poland'),
Row(DEST_COUNTRY_NAME='Portugal'),
Row(DEST_COUNTRY_NAME='Australia'),
```

```
Row(DEST_COUNTRY_NAME='Romania'),
Row(DEST_COUNTRY_NAME='Austria'),
Row(DEST_COUNTRY_NAME='Egypt'),
Row(DEST_COUNTRY_NAME='Costa Rica'),
Row(DEST_COUNTRY_NAME='El Salvador'),
Row(DEST_COUNTRY_NAME='Kazakhstan'),
Row(DEST_COUNTRY_NAME='Burkina Faso'),
Row(DEST_COUNTRY_NAME='South Africa'),
Row(DEST_COUNTRY_NAME='Bermuda'),
Row(DEST_COUNTRY_NAME='Bahrain'),
Row(DEST_COUNTRY_NAME='Colombia'),
Row(DEST_COUNTRY_NAME='Hungary'),
Row(DEST_COUNTRY_NAME='Pakistan'),
Row(DEST_COUNTRY_NAME='United Kingdom'),
Row(DEST_COUNTRY_NAME='Netherlands')]
```

b. List all the unique origin countries. (5%)

[4]:
```
flights_df.select("ORIGIN_COUNTRY_NAME").distinct().collect()
```

[4]:
```
[Row(ORIGIN_COUNTRY_NAME='Paraguay'),
 Row(ORIGIN_COUNTRY_NAME='Russia'),
 Row(ORIGIN_COUNTRY_NAME='Anguilla'),
 Row(ORIGIN_COUNTRY_NAME='Senegal'),
 Row(ORIGIN_COUNTRY_NAME='Sweden'),
 Row(ORIGIN_COUNTRY_NAME='Kiribati'),
 Row(ORIGIN_COUNTRY_NAME='Guyana'),
 Row(ORIGIN_COUNTRY_NAME='Philippines'),
 Row(ORIGIN_COUNTRY_NAME='Fiji'),
 Row(ORIGIN_COUNTRY_NAME='Turkey'),
 Row(ORIGIN_COUNTRY_NAME='Germany'),
 Row(ORIGIN_COUNTRY_NAME='Cambodia'),
 Row(ORIGIN_COUNTRY_NAME='Jordan'),
 Row(ORIGIN_COUNTRY_NAME='Palau'),
 Row(ORIGIN_COUNTRY_NAME='Turks and Caicos Islands'),
 Row(ORIGIN_COUNTRY_NAME='France'),
 Row(ORIGIN_COUNTRY_NAME='Greece'),
 Row(ORIGIN_COUNTRY_NAME='British Virgin Islands'),
 Row(ORIGIN_COUNTRY_NAME='Taiwan'),
 Row(ORIGIN_COUNTRY_NAME='Dominica'),
 Row(ORIGIN_COUNTRY_NAME='Argentina'),
 Row(ORIGIN_COUNTRY_NAME='Angola'),
 Row(ORIGIN_COUNTRY_NAME='Belgium'),
 Row(ORIGIN_COUNTRY_NAME='Congo (Brazaville)'),
 Row(ORIGIN_COUNTRY_NAME='Ecuador'),
 Row(ORIGIN_COUNTRY_NAME='Qatar'),
 Row(ORIGIN_COUNTRY_NAME='Finland'),
```

```
Row(ORIGIN_COUNTRY_NAME='Nicaragua'),
Row(ORIGIN_COUNTRY_NAME='Ghana'),
Row(ORIGIN_COUNTRY_NAME='Peru'),
Row(ORIGIN_COUNTRY_NAME='India'),
Row(ORIGIN_COUNTRY_NAME='United States'),
Row(ORIGIN_COUNTRY_NAME='China'),
Row(ORIGIN_COUNTRY_NAME='Curacao'),
Row(ORIGIN_COUNTRY_NAME='Kuwait'),
Row(ORIGIN_COUNTRY_NAME='Malta'),
Row(ORIGIN_COUNTRY_NAME='Marshall Islands'),
Row(ORIGIN_COUNTRY_NAME='Chile'),
Row(ORIGIN_COUNTRY_NAME='Martinique'),
Row(ORIGIN_COUNTRY_NAME='Cayman Islands'),
Row(ORIGIN_COUNTRY_NAME='Croatia'),
Row(ORIGIN_COUNTRY_NAME='Nigeria'),
Row(ORIGIN_COUNTRY_NAME='Bolivia'),
Row(ORIGIN_COUNTRY_NAME='Italy'),
Row(ORIGIN_COUNTRY_NAME='Suriname'),
Row(ORIGIN_COUNTRY_NAME='Norway'),
Row(ORIGIN_COUNTRY_NAME='Spain'),
Row(ORIGIN_COUNTRY_NAME='Cuba'),
Row(ORIGIN_COUNTRY_NAME='Guadeloupe'),
Row(ORIGIN_COUNTRY_NAME='Denmark'),
Row(ORIGIN_COUNTRY_NAME='Barbados'),
Row(ORIGIN_COUNTRY_NAME='Ireland'),
Row(ORIGIN_COUNTRY_NAME='Morocco'),
Row(ORIGIN_COUNTRY_NAME='Cape Verde'),
Row(ORIGIN_COUNTRY_NAME='Panama'),
Row(ORIGIN_COUNTRY_NAME='Hong Kong'),
Row(ORIGIN_COUNTRY_NAME='Venezuela'),
Row(ORIGIN_COUNTRY_NAME='Ukraine'),
Row(ORIGIN_COUNTRY_NAME='Saint Barthelemy'),
Row(ORIGIN_COUNTRY_NAME='Iceland'),
Row(ORIGIN_COUNTRY_NAME='Israel'),
Row(ORIGIN_COUNTRY_NAME='Saint Kitts and Nevis'),
Row(ORIGIN_COUNTRY_NAME='French Polynesia'),
Row(ORIGIN_COUNTRY_NAME='South Korea'),
Row(ORIGIN_COUNTRY_NAME='Bonaire, Sint Eustatius, and Saba'),
Row(ORIGIN_COUNTRY_NAME='Uruguay'),
Row(ORIGIN_COUNTRY_NAME='Mexico'),
Row(ORIGIN_COUNTRY_NAME='Aruba'),
Row(ORIGIN_COUNTRY_NAME='Indonesia'),
Row(ORIGIN_COUNTRY_NAME='The Bahamas'),
Row(ORIGIN_COUNTRY_NAME='Saint Vincent and the Grenadines'),
Row(ORIGIN_COUNTRY_NAME='Guatemala'),
Row(ORIGIN_COUNTRY_NAME='Azerbaijan'),
Row(ORIGIN_COUNTRY_NAME='Grenada'),
```

```
Row(ORIGIN_COUNTRY_NAME='Sint Maarten'),
Row(ORIGIN_COUNTRY_NAME='Federated States of Micronesia'),
Row(ORIGIN_COUNTRY_NAME='Tunisia'),
Row(ORIGIN_COUNTRY_NAME='Honduras'),
Row(ORIGIN_COUNTRY_NAME='Trinidad and Tobago'),
Row(ORIGIN_COUNTRY_NAME='Saudi Arabia'),
Row(ORIGIN_COUNTRY_NAME='French Guiana'),
Row(ORIGIN_COUNTRY_NAME='Switzerland'),
Row(ORIGIN_COUNTRY_NAME='Ethiopia'),
Row(ORIGIN_COUNTRY_NAME='Jamaica'),
Row(ORIGIN_COUNTRY_NAME='Latvia'),
Row(ORIGIN_COUNTRY_NAME='United Arab Emirates'),
Row(ORIGIN_COUNTRY_NAME='Saint Martin'),
Row(ORIGIN_COUNTRY_NAME='Saint Lucia'),
Row(ORIGIN_COUNTRY_NAME='Canada'),
Row(ORIGIN_COUNTRY_NAME='Samoa'),
Row(ORIGIN_COUNTRY_NAME='Czech Republic'),
Row(ORIGIN_COUNTRY_NAME='Cook Islands'),
Row(ORIGIN_COUNTRY_NAME='Brazil'),
Row(ORIGIN_COUNTRY_NAME='Belize'),
Row(ORIGIN_COUNTRY_NAME='Antigua and Barbuda'),
Row(ORIGIN_COUNTRY_NAME='Dominican Republic'),
Row(ORIGIN_COUNTRY_NAME='Japan'),
Row(ORIGIN_COUNTRY_NAME='Luxembourg'),
Row(ORIGIN_COUNTRY_NAME='New Zealand'),
Row(ORIGIN_COUNTRY_NAME='Greenland'),
Row(ORIGIN_COUNTRY_NAME='Haiti'),
Row(ORIGIN_COUNTRY_NAME='Poland'),
Row(ORIGIN_COUNTRY_NAME='Portugal'),
Row(ORIGIN_COUNTRY_NAME='Australia'),
Row(ORIGIN_COUNTRY_NAME='Romania'),
Row(ORIGIN_COUNTRY_NAME='Bulgaria'),
Row(ORIGIN_COUNTRY_NAME='Austria'),
Row(ORIGIN_COUNTRY_NAME='Costa Rica'),
Row(ORIGIN_COUNTRY_NAME='Egypt'),
Row(ORIGIN_COUNTRY_NAME='Kazakhstan'),
Row(ORIGIN_COUNTRY_NAME='El Salvador'),
Row(ORIGIN_COUNTRY_NAME='South Africa'),
Row(ORIGIN_COUNTRY_NAME='Bermuda'),
Row(ORIGIN_COUNTRY_NAME='Colombia'),
Row(ORIGIN_COUNTRY_NAME='Hungary'),
Row(ORIGIN_COUNTRY_NAME='Pakistan'),
Row(ORIGIN_COUNTRY_NAME='United Kingdom'),
Row(ORIGIN_COUNTRY_NAME='Netherlands')]
```

c. What are the origin and destination countries with maximum count? (10%)

[5]:
```
flights_df.orderBy("count",ascending=False).
 ↪select("DEST_COUNTRY_NAME","ORIGIN_COUNTRY_NAME").show(1)
```

```
+-----------------+-------------------+
|DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|
+-----------------+-------------------+
|    United States|      United States|
+-----------------+-------------------+
only showing top 1 row
```

d. List all the flight records within the same country. (5%)

[6]:
```
flights_df[flights_df.DEST_COUNTRY_NAME==flights_df.ORIGIN_COUNTRY_NAME].
 ↪collect()
```

[6]:
```
[Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='United States',
count=358354)]
```

e. List all the flights to United States ordered by their counts in ascending order. (8%)

[7]:
```
flights_df[flights_df.DEST_COUNTRY_NAME=="United States"].sort("count").
 ↪collect()
```

[7]:
```
[Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saint Martin',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Congo
(Brazaville)', count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Tunisia', count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Kazakhstan',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Hungary', count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ukraine', count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Cambodia',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Bulgaria',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Greenland',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Indonesia',
count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Croatia', count=2),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Malta', count=2),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saint Vincent and
the Grenadines', count=3),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='French Guiana',
count=4),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Azerbaijan',
```

```
count=5),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Egypt', count=11),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ethiopia',
count=11),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Czech Republic',
count=11),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Romania',
count=12),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Angola', count=12),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Pakistan',
count=12),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Cook Islands',
count=12),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Latvia', count=13),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Paraguay',
count=14),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Bolivia',
count=14),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Morocco',
count=15),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ghana', count=15),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Cape Verde',
count=16),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Uruguay',
count=18),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Finland',
count=19),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Greece', count=19),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Kuwait', count=24),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Samoa', count=25),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Suriname',
count=27),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Kiribati',
count=27),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Fiji', count=27),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Senegal',
count=28),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Martinique',
count=32),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='South Africa',
count=32),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Poland', count=33),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Marshall Islands',
count=35),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Anguilla',
count=35),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Dominica',
```

```
count=36),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Palau', count=38),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='French Polynesia',
count=40),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Nigeria',
count=43),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Austria',
count=46),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Grenada',
count=47),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Guadeloupe',
count=47),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saint Barthelemy',
count=53),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Guyana', count=55),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='India', count=62),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Bonaire, Sint
Eustatius, and Saba', count=63),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Jordan', count=64),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Federated States of
Micronesia', count=71),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saudi Arabia',
count=74),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='New Zealand',
count=77),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Curacao',
count=77),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Norway', count=87),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Barbados',
count=89),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Turkey', count=92),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Qatar', count=96),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Sweden',
count=101),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='British Virgin
Islands', count=101),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saint Lucia',
count=109),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Israel',
count=112),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Antigua and
Barbuda', count=112),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Luxembourg',
count=115),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Philippines',
count=116),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Denmark',
```

```
count=116),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Portugal',
count=122),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Saint Kitts and
Nevis', count=123),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Belize',
count=143),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Russia',
count=151),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Argentina',
count=153),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Chile', count=168),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Nicaragua',
count=170),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Trinidad and
Tobago', count=175),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Iceland',
count=177),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Bermuda',
count=190),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Haiti', count=193),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Turks and Caicos
Islands', count=204),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='United Arab
Emirates', count=226),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Belgium',
count=230),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Australia',
count=235),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Taiwan',
count=240),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Venezuela',
count=258),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Cayman Islands',
count=278),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Sint Maarten',
count=290),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ireland',
count=291),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Switzerland',
count=300),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Peru', count=315),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ecuador',
count=326),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Guatemala',
count=327),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Aruba', count=348),
```

```
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Hong Kong',
count=381),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Italy', count=385),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Honduras',
count=412),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Cuba', count=419),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Spain', count=424),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Panama',
count=460),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='El Salvador',
count=486),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Costa Rica',
count=560),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Brazil',
count=578),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Netherlands',
count=702),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Jamaica',
count=714),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='South Korea',
count=754),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='China', count=767),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Colombia',
count=888),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='France',
count=960),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='The Bahamas',
count=991),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Dominican
Republic', count=1282),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Germany',
count=1343),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Japan',
count=1501),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='United Kingdom',
count=1812),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Mexico',
count=6490),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Canada',
count=8177),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='United States',
count=358354)]
```

Question 2: Data Cleaning in Spark

1. Create a Spark DataFrom from the weather-samples.csv file (provided).
2. Write Spark code to answer the following questions.

a. List the columns of the data set and show the information about the columns. (8%)

[8]:
```
weather_df.printSchema()
```

```
root
 |-- number: integer (nullable = true)
 |-- air_pressure_9am: double (nullable = true)
 |-- air_temp_9am: double (nullable = true)
 |-- avg_wind_direction_9am: double (nullable = true)
 |-- avg_wind_speed_9am: double (nullable = true)
 |-- max_wind_direction_9am: double (nullable = true)
 |-- max_wind_speed_9am: double (nullable = true)
 |-- rain_accumulation_9am: double (nullable = true)
 |-- rain_duration_9am: double (nullable = true)
 |-- relative_humidity_9am: double (nullable = true)
 |-- relative_humidity_3pm: double (nullable = true)
```

b. Print summary statistics for all the columns, e.g., using the describe() method. (8%)

[9]:
```
weather_df.describe().show()
```

```
+-------+-----------------+-----------------+------------------+--------------
-------+-----------------+----------------------+------------------+-----------
----------+------------------+--------------------+--------------------+
|summary|           number| air_pressure_9am|      air_temp_9am|avg_wind_direct
ion_9am|avg_wind_speed_9am|max_wind_direction_9am|max_wind_speed_9am|rain_accumu
lation_9am| rain_duration_9am|relative_humidity_9am|relative_humidity_3pm|
+-------+-----------------+-----------------+------------------+--------------
-------+-----------------+----------------------+------------------+-----------
----------+------------------+--------------------+--------------------+
|  count|             1095|             1092|              1090|
1091|              1092|                  1092|              1091|
1089|              1092|                  1095|                1095|
|   mean|            547.0|918.8825513141026| 64.93300141293575|
142.23551070020164| 5.508284242259157|    148.95351796495402| 7.019513529173236|
0.20307895225528005|294.10805227496246|   34.241402059256586|
35.34472714823471|
| stddev|316.24357700987383|3.184161181422828|11.175514003266809|
69.13785928883635| 4.552813465529014|     67.23801294593558| 5.598209170789135|
1.593952125356949|1598.0787786596147|   25.472066802254194|
22.524079453607285|
|    min|                0|           907.99|             36.752|
15.5|         0.6934514|                  28.9|         1.1855782|
0.0|               0.0|                 6.09|                 5.3|
|    max|             1094|           929.32|             98.906|
343.4|        23.5549782|                 312.2|        29.8407796|
24.02|           17704.0|                92.62|                92.25|
+-------+-----------------+-----------------+------------------+--------------
```

```
-------+-----------------+---------------------+-----------------+-----------
----------+------------------+--------------------+--------------------+
```

c. Print the summary statistics for one column: air_pressure_9am. (5%)

[10]:
```
weather_df.select("air_pressure_9am").describe().show()
```

```
+-------+-----------------+
|summary| air_pressure_9am|
+-------+-----------------+
|  count|             1092|
|   mean|918.8825513141026|
| stddev|3.184161181422828|
|    min|           907.99|
|    max|           929.32|
+-------+-----------------+
```

d. Drop rows with missing values in the air_pressure_9am column. (8%)

[11]:
```
weather_df.select("air_pressure_9am").dropna().show()
```

```
+----------------+
|air_pressure_9am|
+----------------+
|          918.06|
|     917.3476881|
|          923.04|
|     920.5027512|
|          921.16|
|           915.3|
|     915.5988675|
|          918.07|
|          920.08|
|          915.01|
|          919.65|
|          915.64|
|          917.39|
|          920.82|
|           911.0|
|     922.3831312|
|          917.89|
|     916.9152554|
|           918.8|
|          922.04|
+----------------+
only showing top 20 rows
```

e. How many rows are dropped at the previous step? (5%)

[12]:
```
print("There are {} rows dropped at the previous step".format(weather_df.
 ↪count()-weather_df.select("air_pressure_9am").dropna().count()))
```

```
There are 3 rows dropped at the previous step
```

f. What is the difference between the mean values of air_temp_9am before and after dropping all the missing values? (5%)

[13]:
```
before_mean = weather_df.select("air_temp_9am").groupBy().mean().
 ↪first()["avg(air_temp_9am)"]
after_mean = weather_df.dropna().select("air_temp_9am").groupBy().mean().
 ↪first()["avg(air_temp_9am)"]
print("Before: {}".format(before_mean))
print("After: {}".format(after_mean))
print("The difference between the mean values of air_temp_9am before and after␣
 ↪dropping all the missing values is {}.".format(after_mean-before_mean))
```

```
Before: 64.93300141293575
After: 65.02260949566728
The difference between the mean values of air_temp_9am before and after dropping
all the missing values is 0.0896080827315302.
```

g. Compute correlation between two columns: rain_accumulation_9am and rain_duration_9am. (5%)

[14]:
```
print("The correlation between two columns: rain_accumulation_9am and␣
 ↪rain_duration_9am is {}.".format(weather_df.
 ↪corr("rain_accumulation_9am","rain_duration_9am")))
```

```
The correlation between two columns: rain_accumulation_9am and rain_duration_9am
is 0.7337968783308563.
```

0.1.1 Impute missing values.

h. Instead of removing rows containing missing values, replace the values with the mean value for that column. First, load the avg function and make a copy of the original DataFrame. (5%)

[15]:
```
from pyspark.sql.functions import avg
weather_df_impute = weather_df
```

i. Next, iterate through each column in the DataFrame, compute the mean value for that column and then replace any missing values in that column with the mean. (10%)

[16]:
```
mean = weather_df_impute.agg(*(avg(c).alias(c) for c in weather_df_impute.
 ↪columns))
meaninfo = mean.first().asDict()
weather_df_impute = weather_df_impute.fillna(meaninfo)
```

j. Print imputed data summary statistics. Call describe() to show the summary statistics for the original and imputed air_temp_9am. What is the difference between the mean values of air_temp_9am before and after the imputation? (8%)

[17]:
```
before_mean = weather_df_impute.groupBy().mean().first()["avg(air_temp_9am)"]
after_mean =  weather_df_impute.groupBy().mean().first()["avg(air_temp_9am)"]
print("Before: {}".format(before_mean))
print("After: {}".format(after_mean))
print("The difference between the mean values of air_temp_9am before and after␣
 ↪the imputation is {}.".format(after_mean-before_mean))
```

```
Before: 64.93300141293575
After: 64.93300141293575
The difference between the mean values of air_temp_9am before and after the
imputation is 0.0.
```

[18]:
```
spark.stop()
```