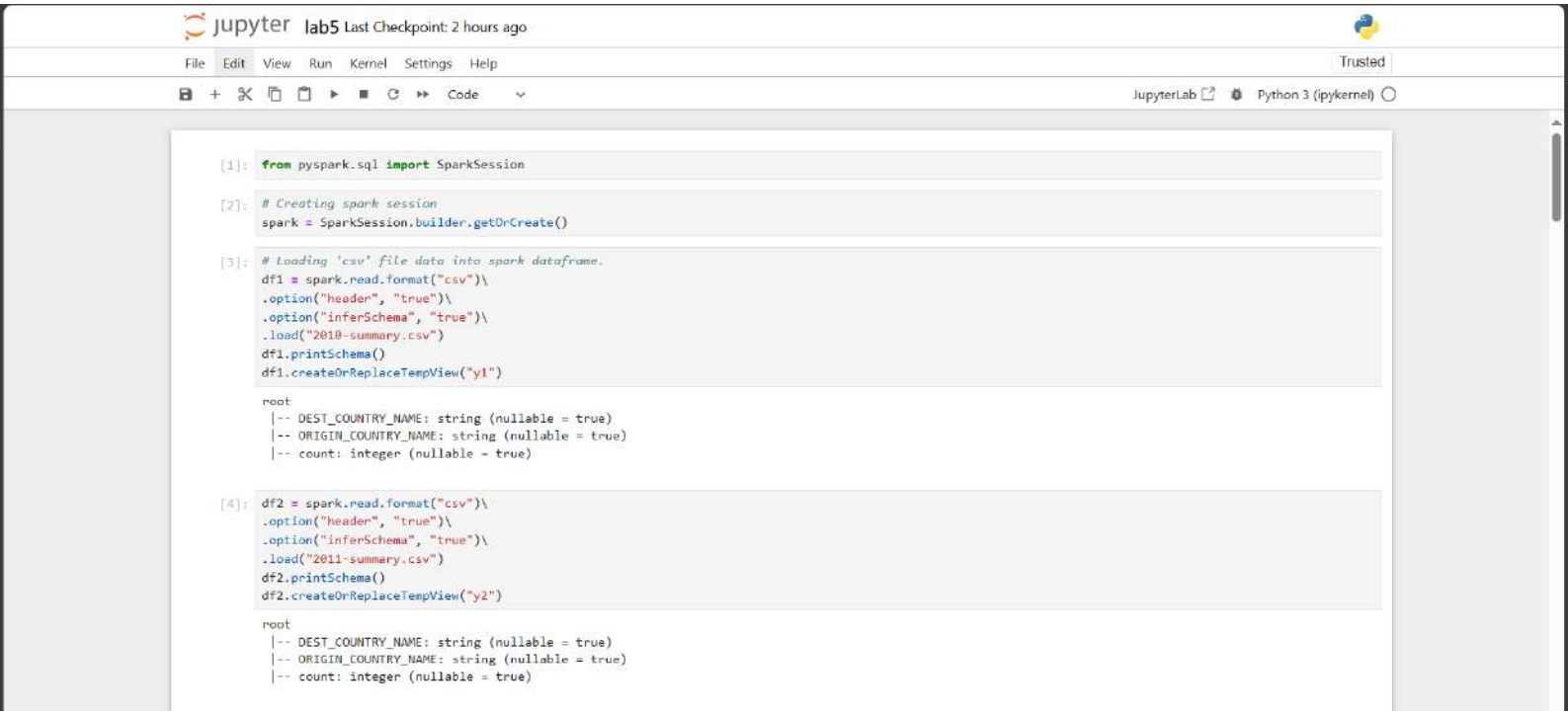
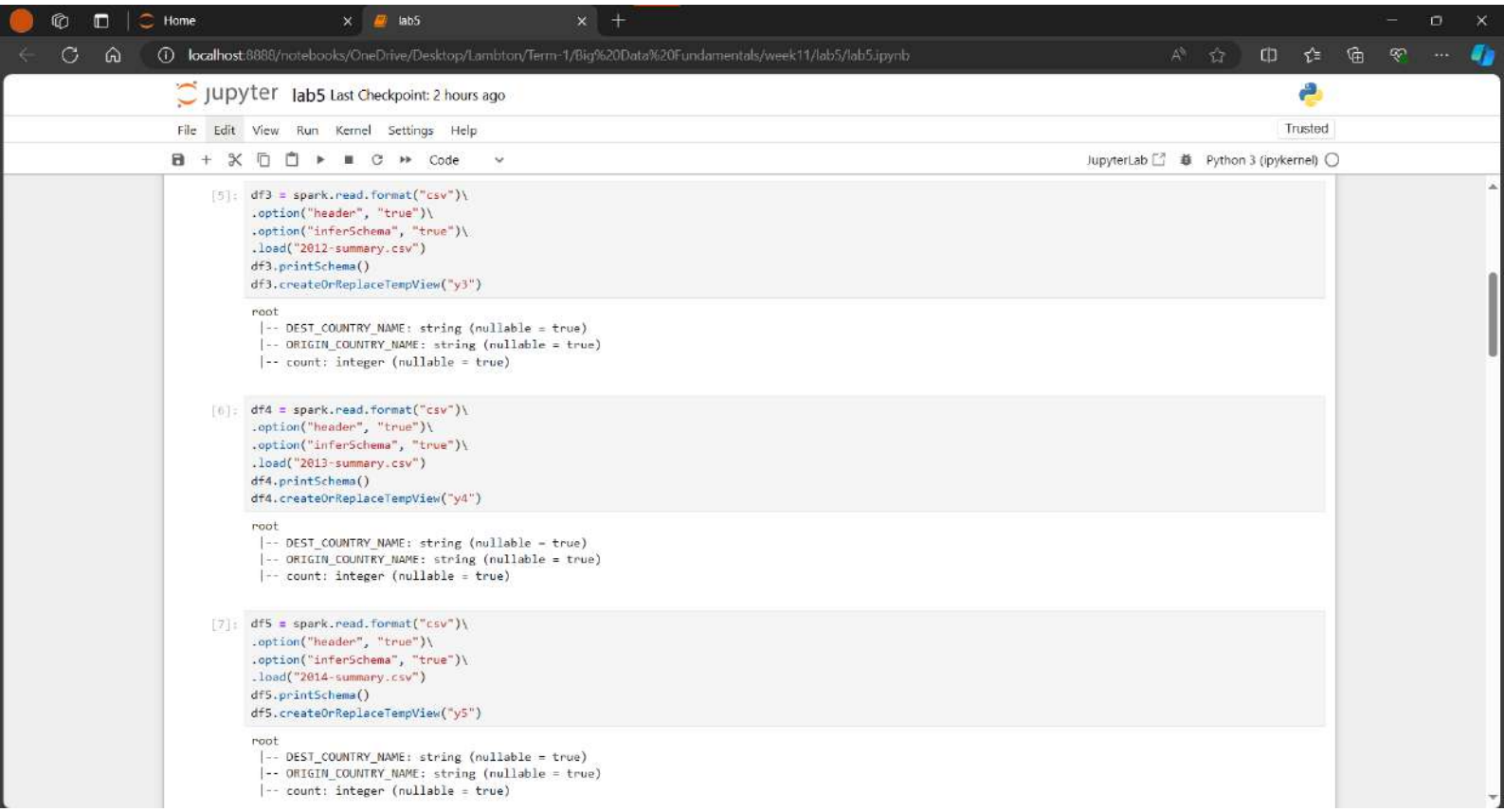
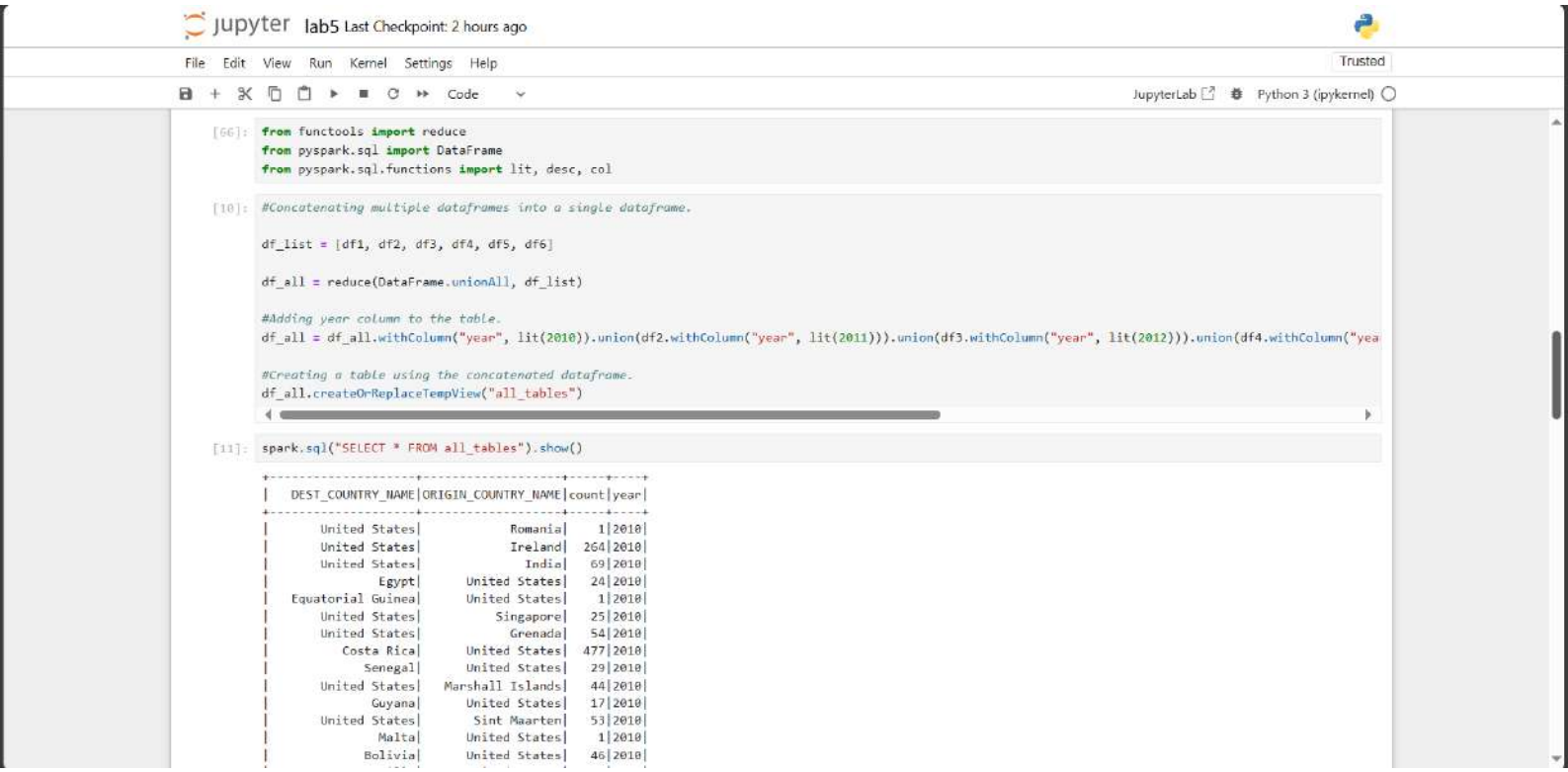
1. Created six different dataframes using six csv files.



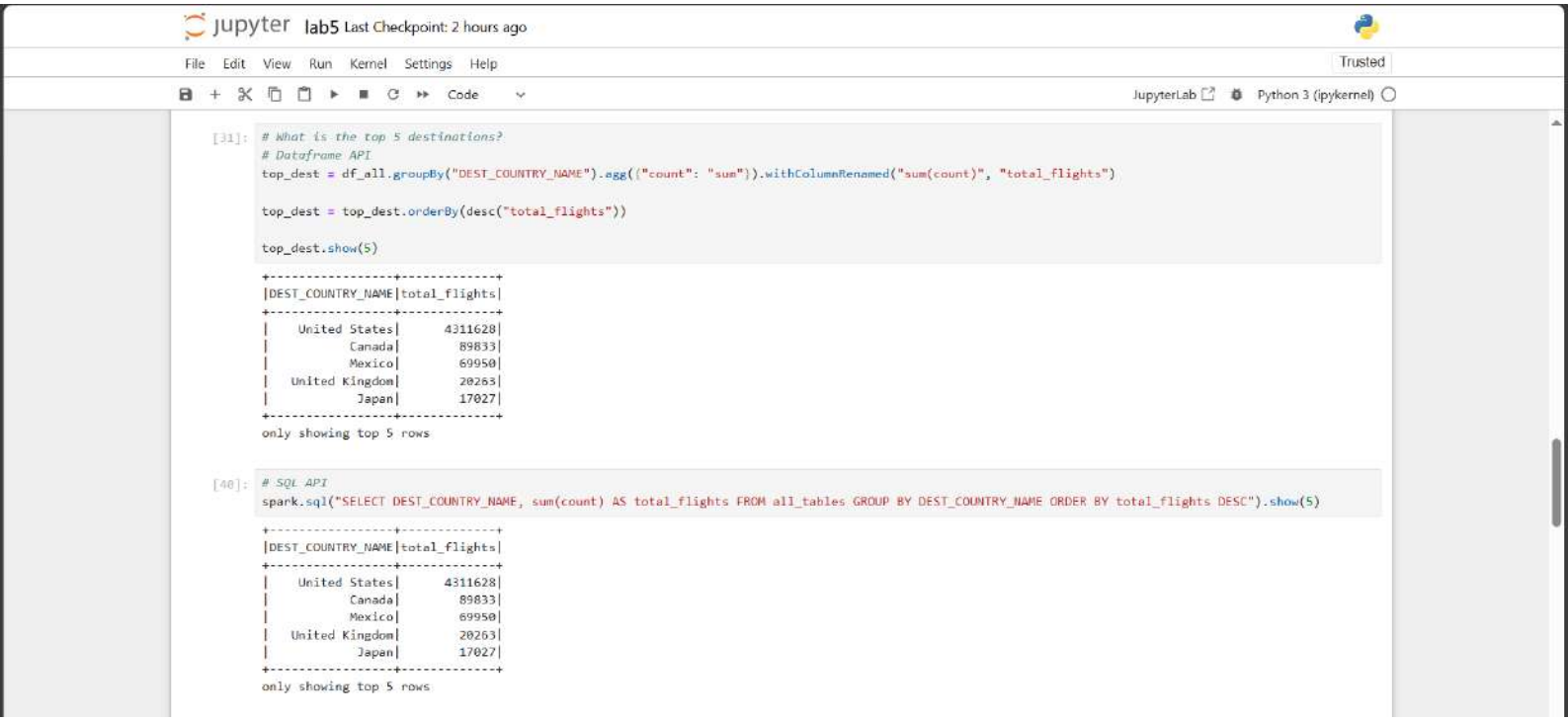

Home
lab5
localhost:8888/notebooks/OneDrive/Desktop/Lambton/Term-1/Big%20Data%20Fundamentals/week11/lab5/lab5.ipynb
jupyter lab5 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[5]:
df3 = spark.read.format("csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("2012-summary.csv")
df3.printSchema()
df3.createOrReplaceTempView("y3")
root
|-- DEST_COUNTRY_NAME: string (nullable = true)
|-- ORIGIN_COUNTRY_NAME: string (nullable = true)
|-- count: integer (nullable = true)
[6]:
df4 = spark.read.format("csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("2013-summary.csv")
df4.printSchema()
df4.createOrReplaceTempView("y4")
root
|-- DEST_COUNTRY_NAME: string (nullable = true)
|-- ORIGIN_COUNTRY_NAME: string (nullable = true)
|-- count: integer (nullable = true)
[7]:
df5 = spark.read.format("csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("2014-summary.csv")
df5.printSchema()
df5.createOrReplaceTempView("y5")
root
|-- DEST_COUNTRY_NAME: string (nullable = true)
|-- ORIGIN_COUNTRY_NAME: string (nullable = true)
|-- count: integer (nullable = true)

jupyter lab5 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[8]:
df6 = spark.read.format("csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("2015-summary.csv")
df6.printSchema()
df6.createOrReplaceTempView("y6")
root
|-- DEST_COUNTRY_NAME: string (nullable = true)
|-- ORIGIN_COUNTRY_NAME: string (nullable = true)
|-- count: integer (nullable = true)

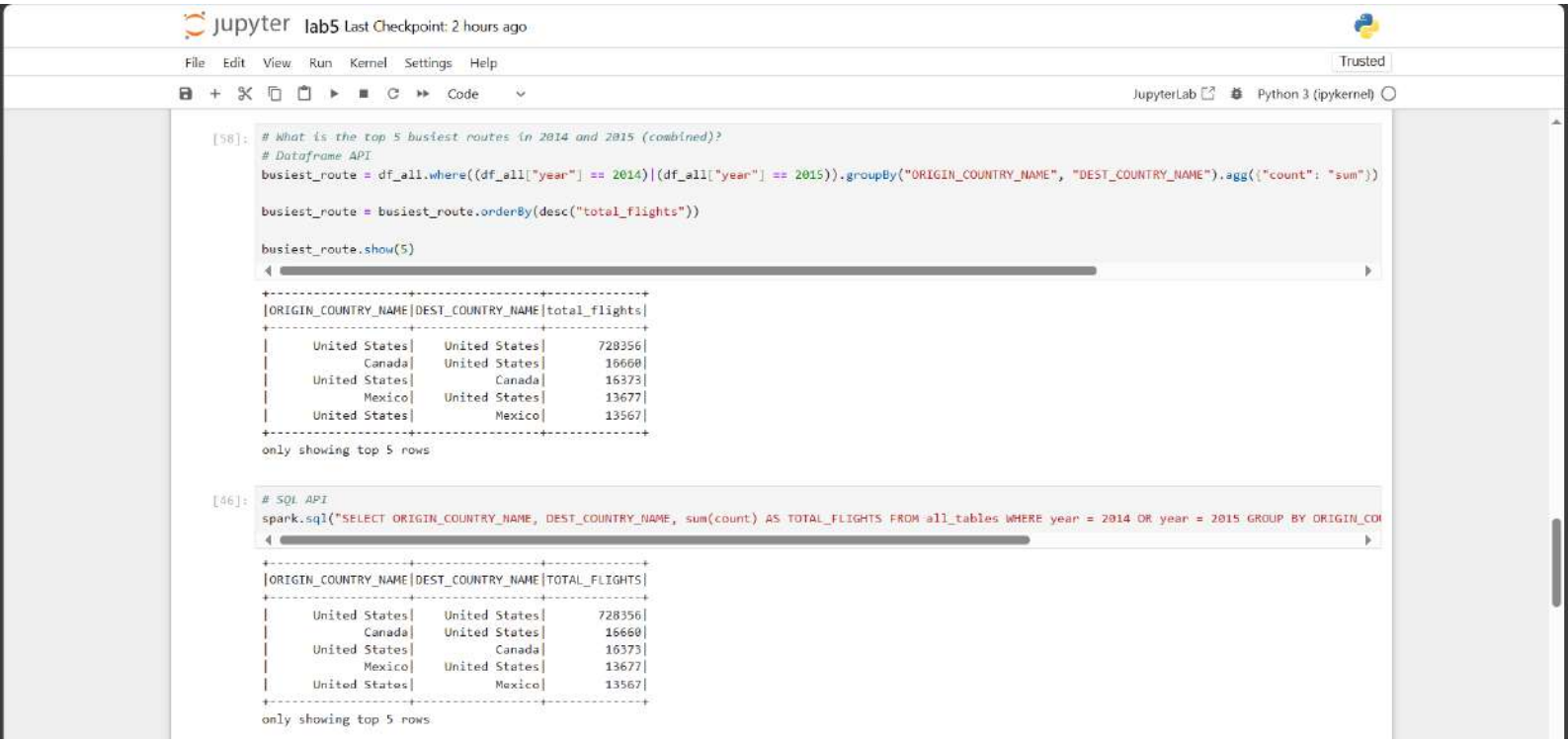
2. Importing necessary modules and Concatenate all six DataFrames into one single DataFrame.



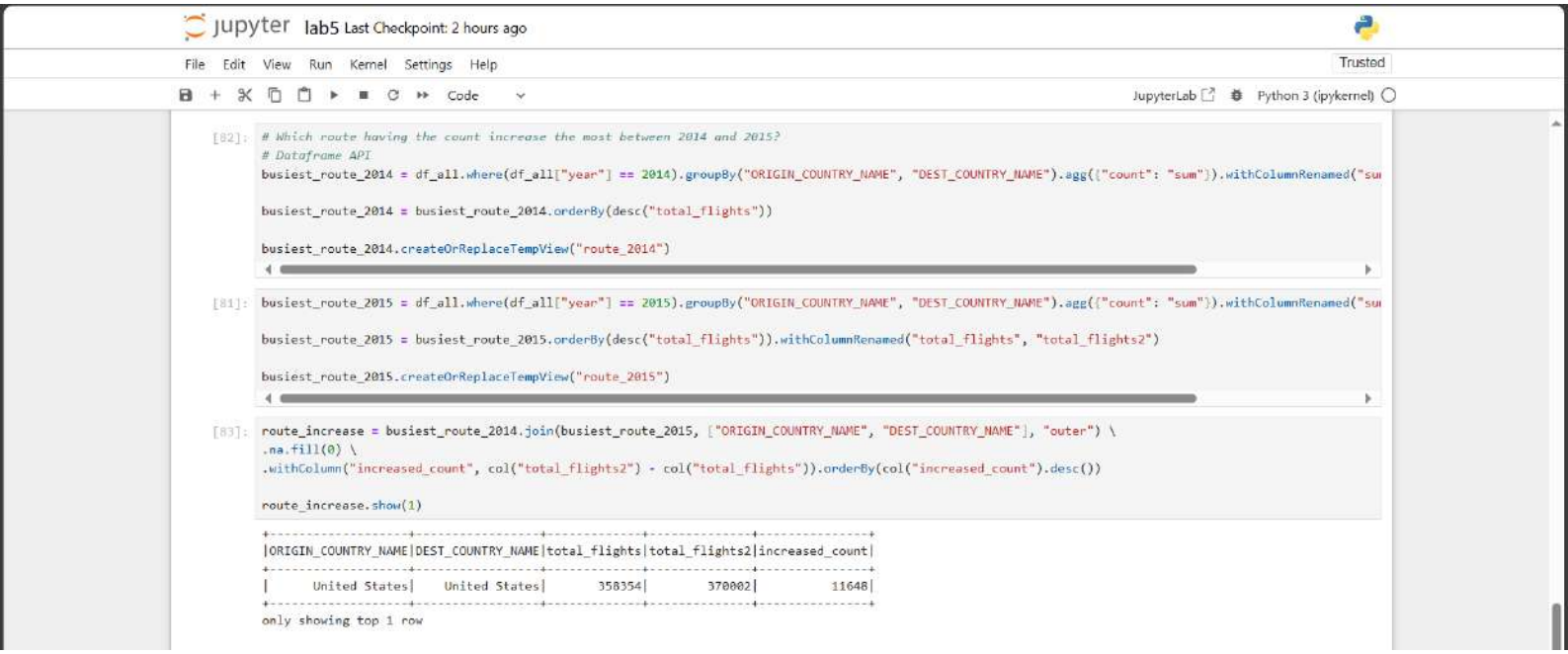
3. Top 5 Destinations.



4. Top 5 busiest routes in 2014 and 2015 (combined)



5. Route having the count increase the most between 2014 and 2015



jupyter lab5 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[89]:
SQL API
spark.sql("""SELECT r1.ORIGIN_COUNTRY_NAME, r1.DEST_COUNTRY_NAME, (r2.total_flights2 - r1.total_flights) AS increased_count
FROM route_2014 r1
JOIN route_2015 r2
ON r1.ORIGIN_COUNTRY_NAME = r2.ORIGIN_COUNTRY_NAME
AND r1.DEST_COUNTRY_NAME = r2.DEST_COUNTRY_NAME
ORDER BY increased_count DESC
LIMIT 1""").show()
+-----------------+-----------------+--------------+
|ORIGIN_COUNTRY_NAME|DEST_COUNTRY_NAME|increased_count|
+-----------------+-----------------+--------------+
| United States| United States| 11648|
+-----------------+-----------------+--------------+