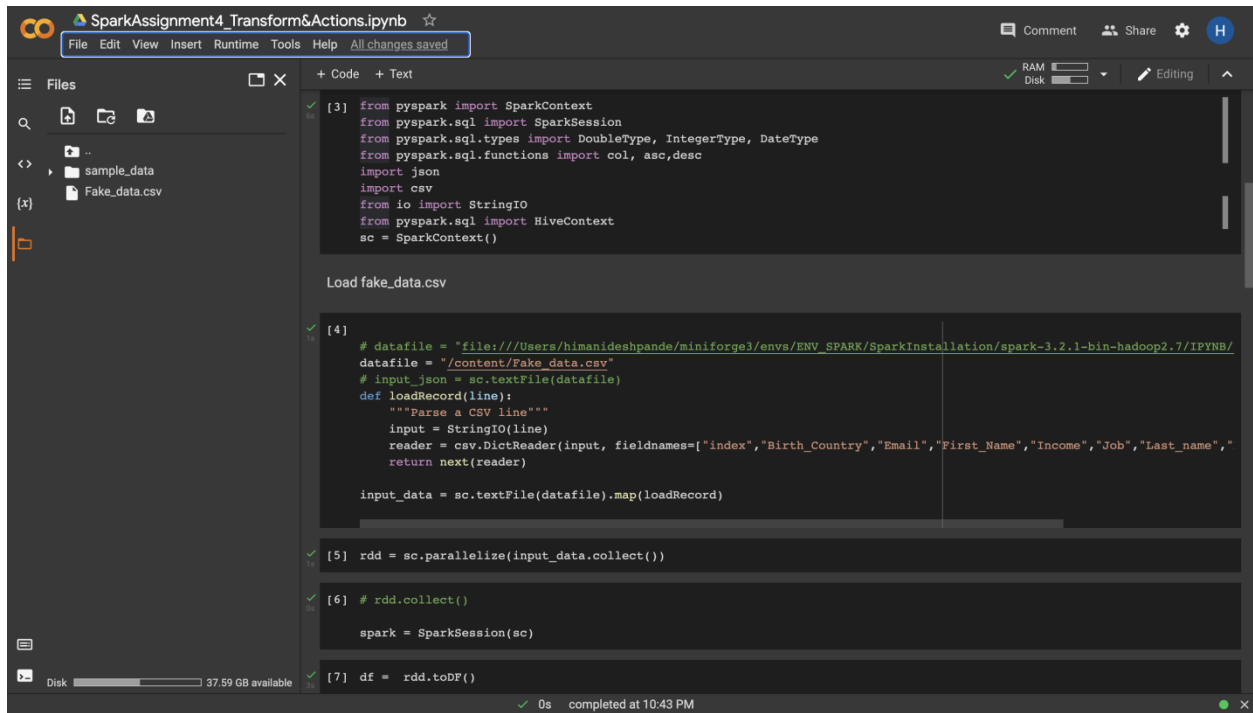


Assignment 4

By Himani Anil Deshpande

Load fake_data.csv



The screenshot shows a Jupyter Notebook titled "SparkAssignment4_Transform&Actions.ipynb". The left sidebar displays a file explorer with a folder named "sample_data" containing a file "Fake_data.csv". The main code area contains the following code blocks:

```
[3] from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.sql.types import DoubleType, IntegerType, DateType
from pyspark.sql.functions import col, asc, desc
import json
import csv
from io import StringIO
from pyspark.sql import HiveContext
sc = SparkContext()
```

Load fake_data.csv

```
[4] # datafile = "file:///Users/himanideshpande/miniforge3/envs/ENV_SPARK/SparkInstallation/spark-3.2.1-bin-hadoop2.7/IPYNB/
datafile = "/content/Fake_data.csv"
# input_json = sc.textFile(datafile)
def loadRecord(line):
    """Parse a CSV line"""
    input = StringIO(line)
    reader = csv.DictReader(input, fieldnames=["index", "Birth_Country", "Email", "First_Name", "Income", "Job", "Last_name", "
    return next(reader)

input_data = sc.textFile(datafile).map(loadRecord)
```

```
[5] rdd = sc.parallelize(input_data.collect())
```

```
[6] # rdd.collect()

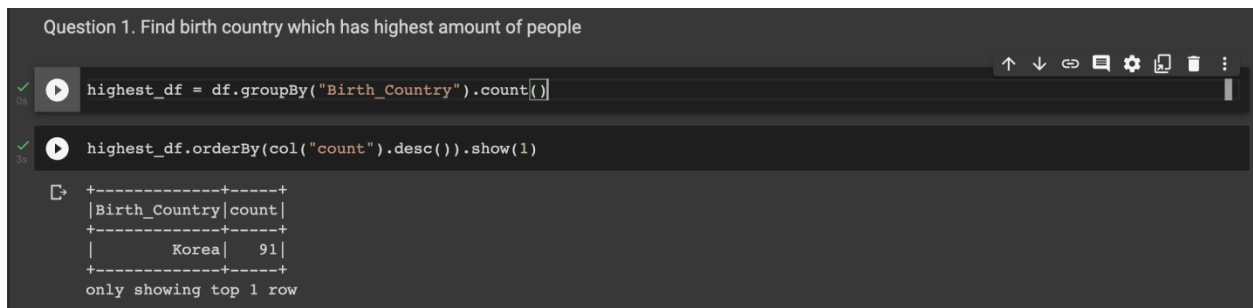
spark = SparkSession(sc)
```

```
[7] df = rdd.toDF()
```

The notebook interface shows a status bar at the bottom indicating "0s completed at 10:43 PM".

1) Find birth country which has highest amount of people

Using Transformations and Actions



The screenshot shows a Jupyter Notebook titled "Question 1. Find birth country which has highest amount of people". The code area contains the following code blocks:

```
highest_df = df.groupBy("Birth_Country").count()
```

```
highest_df.orderBy(col("count").desc()).show(1)
```

The output of the code is displayed below the code blocks:

```
+-----+-----+
|Birth_Country|count|
+-----+-----+
|      Korea|    91|
+-----+-----+
only showing top 1 row
```

Using Spark SQL

Question 1. Find birth country which has highest amount of people

```
teOrReplaceTempView("Fake_data")

spark.sql("SELECT COUNT(Birth_Country), Birth_Country FROM Fake_data group By Birth_Country order by 1 DESC Limit 1")
show()
```

count(Birth_Country)	Birth_Country
91	Korea

2) Find average income of people who are born in **united states of america**

Using Transformations and Actions

Question 2 Find average income of people who are born in united states of america

```
[14] df.filter(df['Birth_Country'] == 'United States of America').groupBy('Birth_Country').avg('Income').show()
```

Birth_Country	avg(Income)
United States of ...	208759.82352941178

Using Spark SQL

Question 2 Find average income of people who are born in united states of america

```
l("SELECT Avg(tb.Income), tb.Birth_Country FROM Fake_data tb group by tb.Birth_Country having tb.Birth_Country = 'United States of America'")
show()
```

avg(Income)	Birth_Country
208759.82352941178	United States of ...

3) How many people has income over 100,000 but their loan is not approved.

Using Transformations and Actions

Question 3 How many people has income over 100,000 but their loan is not approved.

```
[15] df_count_income_high = df.filter((df.Income > 100000) & (df.Loan_Approved == False))

df_count_income_high.count()

4009
```

Using Spark SQL

Question 3 How many people has income over 100,000 but their loan is not approved.

```
[ ]
sqlDF = spark.sql("SELECT COUNT(*) FROM Fake_data tb where tb.Income > 100000 and tb.Loan_Approved = False ")
sqlDF.show()

+-----+
|count(1)|
+-----+
|      4009|
+-----+
```

4) Find top 10 people with highest income in **United States of america**. (Print their names, income and jobs) – 10

Using Transformations and Actions

Question 4 Find top 10 people with highest income in United States of america. (Print their names, income and jobs)

```
[23] top_paid = df.filter(df['Birth_Country'] == 'United States of America').orderBy(col("Income").desc())

[24] top_paid.select("First_Name", "Last_name", "Job", "Income").show(10)

+-----+-----+-----+-----+
|First_Name|Last_name|Job|Income|
+-----+-----+-----+-----+
|Alyssa|Miller|Amenity horticult...|482588|
|Hunter|Walls|Psychologist, pri...|468946|
|Rose|Henderson|Adult guidance wo...|426115|
|Danielle|Leonard|Furniture conserv...|389810|
|Terry|Klein|Meteorologist|380410|
|Cindy|Newton|Research scientis...|370322|
|Scott|Mitchell|Art therapist|368913|
|Christy|Sandoval|Engineer, land|355150|
|Kelly|Reynolds|Press sub|341448|
|Kristina|Smith|Herbalist|338804|
+-----+-----+-----+-----+
only showing top 10 rows
```

Using Spark SQL

Question 4 Find top 10 people with highest income in United States of america. (Print their names, income and jobs)

```
[ ]
sqlDF = spark.sql("SELECT tb.First_Name,tb.Last_name, tb.Job, tb.Income FROM Fake_data tb where tb.Birth_Country = 'Uni
sqlDF.show()
```

```
+-----+-----+-----+-----+
|First_Name|Last_name|Job|Income|
+-----+-----+-----+-----+
|Alyssa|Miller|Amenity horticult...|482588|
|Hunter|Walls|Psychologist, pri...|468946|
|Rose|Henderson|Adult guidance wo...|426115|
|Danielle|Leonard|Furniture conserv...|389810|
|Terry|Klein|Meteorologist|380410|
|Cindy|Newton|Research scientis...|370322|
|Scott|Mitchell|Art therapist|368913|
|Christy|Sandoval|Engineer, land|355150|
|Kelly|Reynolds|Press sub|341448|
|Kristina|Smith|Herbalist|338804|
+-----+-----+-----+-----+
```

5) How many number of distinct jobs are there?

Using Transformations and Actions

Question 5 How many number of distinct jobs are there?

```
✓ 0s df.select("Job").distinct().count()
```

640

Using Spark SQL

Question 5 How many number of distinct jobs are there?

```
▶
sqlDF = spark.sql("SELECT count(DISTINCT(tb.Job)) FROM Fake_data tb ")
sqlDF.show()
```

```
▶
+-----+
|count(DISTINCT Job)|
+-----+
|640|
+-----+
```

6) How many writers earn less than 100,000?

Using Transformations and Actions

Question 6 How many writers earn less than 100,000?

```
✓ 0s df.filter((df.Income <100000) & (df.Job == 'Writer')).count()
```

5

+ Code + Text

Using Spark SQL

Question 6 How many writers earn less than 100,000?



```
sqlDF = spark.sql("SELECT count(*) FROM Fake_data tb where tb.Income <100000 and tb.Job = 'Writer' ")
sqlDF.show()
```



```
+-----+
|count(1)|
+-----+
|        5|
+-----+
```

References:

<https://www.datasciencemadesimple.com/distinct-value-of-a-column-in-pyspark/#:~:text=Distinct%20value%20of%20the%20column%20in%20pyspark%20is%20obtained%20by,value%20of%20those%20columns%20combined>.

<https://spark.apache.org/docs/2.2.0/sql-programming-guide.html#:~:text=Spark%20SQL%20is%20a%20Spark,information%20to%20perform%20extra%20optimizations>