You are working on a data analysis project at Deloitte where you need to analyze a dataset containing information about various cities. Your task is to calculate the population density of these cities, rounded to the nearest integer, and identify the cities with the minimum and maximum densities. The population density should be calculated as (Population / Area).

The output should contain 'city', 'country', 'density'.

**city:**varchar **country:**varchar **population:**int **area:**float

| **city** | **country** | **population** | **area** |
| --- | --- | --- | --- |
| Metropolis | Countryland | 1000000 | 500 |
| Smallville | Countryland | 50000 | 1000 |
| Coastcity | Oceanland | 300000 | 0 |
| Starcity | Mountainous | 600000 | 600 |
| Gotham | Islander | 1500000 | 300 |
| Rivertown | Plainsland | 100000 | 5000 |
| Lakecity | Forestland | 100000 | 5000 |
| Hilltown | Hillside | 200000 | 450 |
| Forestville | Forestland | 500000 | 700 |
| Oceanview | Seaside | 800000 | 0 |

**Solution:**

# Import your libraries

import pyspark

from pyspark.sql.functions import \*

#removing the records that has 'area' as '0.0'

filter\_df=cities\_population.where(col('area')!= '0.0' )

# Start writing code

density\_df=filter\_df.withColumn('Density',round(col('population')/col('area')))

max\_df=density\_df.orderBy(col('Density').desc()).limit(1)

min\_df=density\_df.orderBy(col('Density').asc()).limit(1)

final\_df=max\_df.union(min\_df).select('city','country','density')

# To validate your solution, convert your final pySpark df to a pandas df

final\_df.toPandas()

**Output**

View the output in a separate browser tab

**Execution time:**0.37207 seconds

| **city** | **country** | **density** |
| --- | --- | --- |
| Gotham | Islander | 5000 |
| Rivertown | Plainsland | 20 |