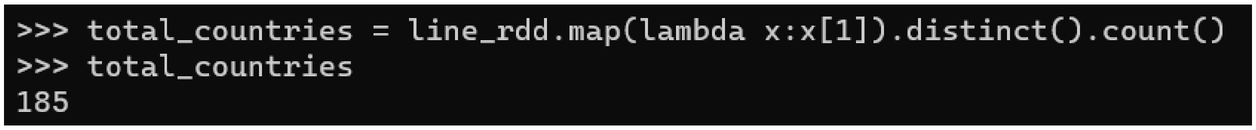
file_rdd = sc.textFile('hdfs://localhost:9000/user/5000_Sales_Records.csv',2)
line_rdd = file_rdd.map(lambda x:x.split(','))
line_rdd = line_rdd.filter(lambda x:x[0]!='Region')
line_rdd.cache()

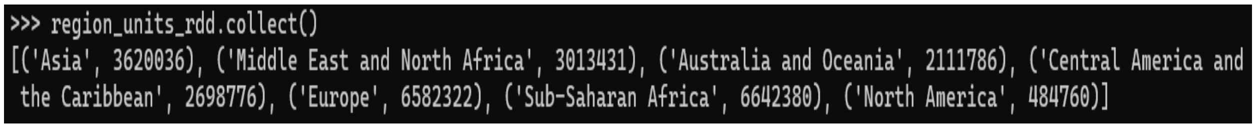
Query questions

1. Display the number of countries present in the data(Using Hive)



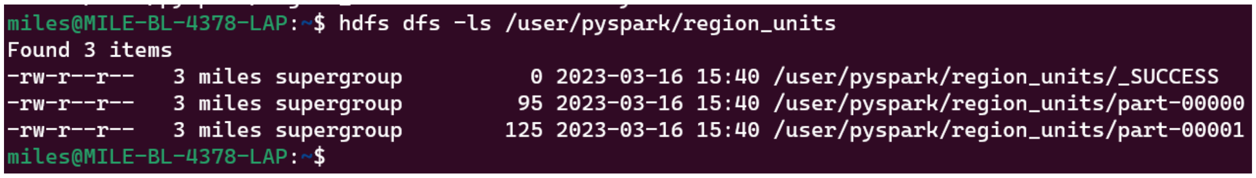
```
>>> total_countries = line_rdd.map(lambda x:x[1]).distinct().count()
>>> total_countries
185
```

2.Display the number of units sold in each region.(Using Hive)
region_units_rdd = line_rdd.map(lambda x:(x[0],x[8])).reduceByKey(lambda a,b:int(a)+int(b))



```
>>> region_units_rdd.collect()
[('Asia', 3620036), ('Middle East and North Africa', 3013431), ('Australia and Oceania', 2111786), ('Central America and
 the Caribbean', 2698776), ('Europe', 6582322), ('Sub-Saharan Africa', 6642380), ('North America', 484760)]
```

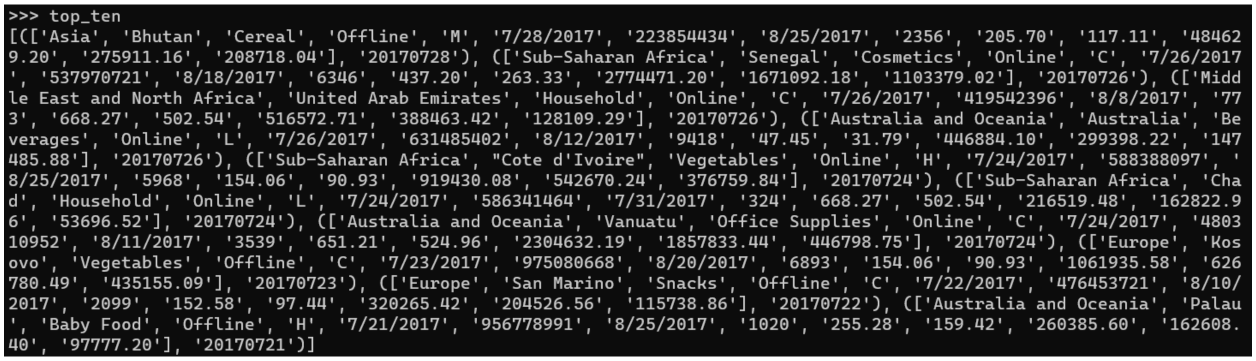
region_units_rdd.saveAsTextFile("hdfs://localhost:9000/user/pyspark/region_units")



```
miles@MILE-BL-4378-LAP:~$ hdfs dfs -ls /user/pyspark/region_units
Found 3 items
-rw-r--r--   3 miles supergroup          0 2023-03-16 15:40 /user/pyspark/region_units/_SUCCESS
-rw-r--r--   3 miles supergroup         95 2023-03-16 15:40 /user/pyspark/region_units/part-00000
-rw-r--r--   3 miles supergroup        125 2023-03-16 15:40 /user/pyspark/region_units/part-00001
miles@MILE-BL-4378-LAP:~$
```

3.Display the 10 most recent sales. (Using Hive)
#Converting date format, splitting the date on ‘/’ and order it as YYYYmmdd (If single digit then added 0 in front) & convert it into int

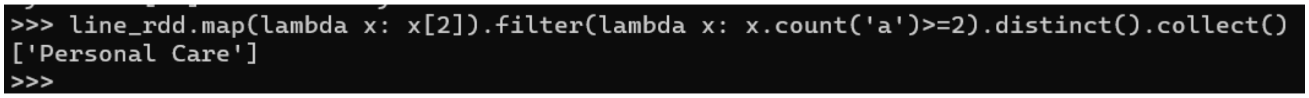
```
def date_format(date):
    split_date = date.split('/')
    if len(split_date[0]) == 1:
        split_date[0] = '0'+split_date[0]
    if len(split_date[1]) == 1:
        split_date[1] = '0'+split_date[1]
    date = split_date[2]+split_date[0]+split_date[1]
    return date
date_rdd = line_rdd.map(lambda x:(x,date_format(x[5])))
top_ten_rdd = date_rdd.sortBy(lambda x:x[1],ascending=False)
top_ten = top_ten_rdd.take(10)
```



```
>>> top_ten
[(['Asia', 'Bhutan', 'Cereal', 'Offline', 'M', '7/28/2017', '223854434', '8/25/2017', '2356', '205.70', '117.11', '48462
9.20', '275911.16', '208718.04'], '20170728'), (['Sub-Saharan Africa', 'Senegal', 'Cosmetics', 'Online', 'C', '7/26/2017
', '537970721', '8/18/2017', '6346', '437.20', '263.33', '2774471.20', '1671092.18', '1103379.02'], '20170726'), (['Midd
le East and North Africa', 'United Arab Emirates', 'Household', 'Online', 'C', '7/26/2017', '419542396', '8/8/2017', '77
3', '668.27', '502.54', '516572.71', '388463.42', '128109.29'], '20170726'), (['Australia and Oceania', 'Australia', 'Be
verages', 'Online', 'L', '7/26/2017', '631485402', '8/12/2017', '9418', '47.45', '31.79', '446884.10', '299398.22', '147
485.88'], '20170726'), (['Sub-Saharan Africa', "Cote d'Ivoire", 'Vegetables', 'Online', 'H', '7/24/2017', '588388097', '
8/25/2017', '5968', '154.06', '90.93', '919430.08', '542670.24', '376759.84'], '20170724'), (['Sub-Saharan Africa', 'Cha
d', 'Household', 'Online', 'L', '7/24/2017', '586341464', '7/31/2017', '324', '668.27', '502.54', '216519.48', '162822.9
6', '53696.52'], '20170724'), (['Australia and Oceania', 'Vanuatu', 'Office Supplies', 'Online', 'C', '7/24/2017', '4803
10952', '8/11/2017', '3539', '651.21', '524.96', '2304632.19', '1857833.44', '446798.75'], '20170724'), (['Europe', 'Kos
ovo', 'Vegetables', 'Offline', 'C', '7/23/2017', '975080668', '8/20/2017', '6893', '154.06', '90.93', '1061935.58', '626
780.49', '435155.09'], '20170723'), (['Europe', 'San Marino', 'Snacks', 'Offline', 'C', '7/22/2017', '476453721', '8/10/
2017', '2099', '152.58', '97.44', '320265.42', '204526.56', '115738.86'], '20170722'), (['Australia and Oceania', 'Palau
', 'Baby Food', 'Offline', 'H', '7/21/2017', '956778991', '8/25/2017', '1020', '255.28', '159.42', '260385.60', '162608.
40', '97777.20'], '20170721')]
```

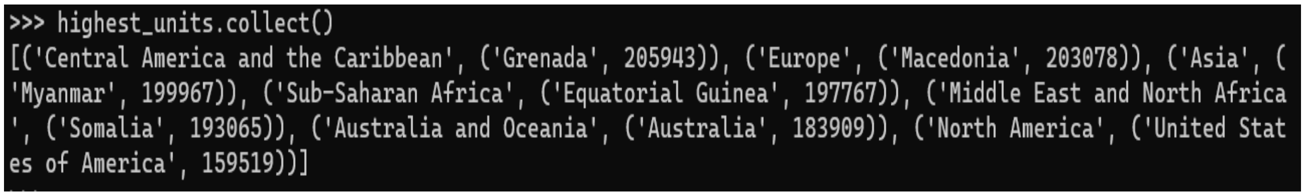
top_sales_rdd = sc.parallelize(top_ten,1)
top_sales_rdd.saveAsTextFile("hdfs://localhost:9000/user/pyspark/top_ten_sales")

4. Display the products with atleast 2 occurences of 'a' (Using spark)



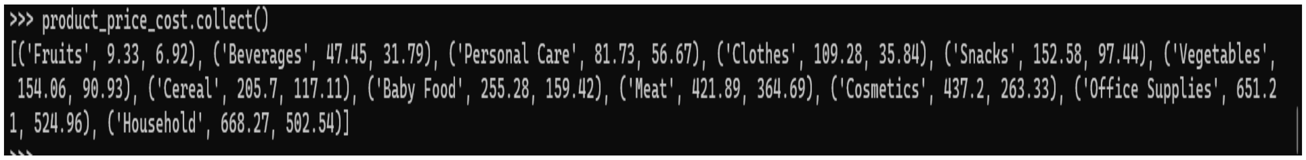
```
>>> line_rdd.map(lambda x: x[2]).filter(lambda x: x.count('a')>=2).distinct().collect()
['Personal Care']
>>>
```

5.Display country in each region with highest units sold. (Using spark)
country_units_rdd = line_rdd.map(lambda x:((x[0],x[1]),x[8]))
country_sum_units = country_units_rdd.reduceByKey(lambda a,b:a+b)
country_units_sorted = country_sum_units.sortBy(lambda x:x[1],ascending = False)
highest_units = country_units_sorted.map(lambda x:(x[0][0],(x[0][1],x[1]))).reduceByKey(lambda a,b:max(a,b,key = lambda x:x[1])).sortBy(lambda x:x[1][1],ascending=False)



```
>>> highest_units.collect()
[('Central America and the Caribbean', ('Grenada', 205943)), ('Europe', ('Macedonia', 203078)), ('Asia', (
'Myanmar', 199967)), ('Sub-Saharan Africa', ('Equatorial Guinea', 197767)), ('Middle East and North Africa
', ('Somalia', 193065)), ('Australia and Oceania', ('Australia', 183909)), ('North America', ('United Stat
es of America', 159519))]
```

6.Display the unit price and unit cost of each item in ascending order. (Using spark)
data_rdd = line_rdd.map(lambda x:(x[2],float(x[9]),float(x[10])))
product_price_cost = data_rdd.distinct().sortBy(lambda x:(x[1],x[2]))



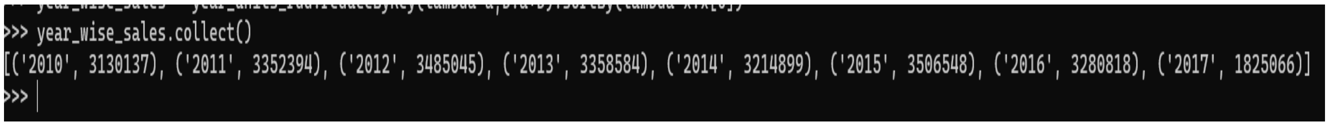
```
>>> product_price_cost.collect()
[('Fruits', 9.33, 6.92), ('Beverages', 47.45, 31.79), ('Personal Care', 81.73, 56.67), ('Clothes', 109.28, 35.84), ('Snacks', 152.58, 97.44), ('Vegetables',
 154.06, 90.93), ('Cereal', 205.7, 117.11), ('Baby Food', 255.28, 159.42), ('Meat', 421.89, 364.69), ('Cosmetics', 437.2, 263.33), ('Office Supplies', 651.2
1, 524.96), ('Household', 668.27, 502.54)]
```

product_price_cost.saveAsTextFile("hdfs://localhost:9000/user/pyspark/product_price_cost")

7.Display the number of sales yearwise. (Using pyspark)

year_units_rdd = line_rdd.map(lambda x:(x[5][-4:],int(x[8])))

year_wise_sales = year_units_rdd.reduceByKey(lambda a,b:a+b).sortBy(lambda x:x[0])

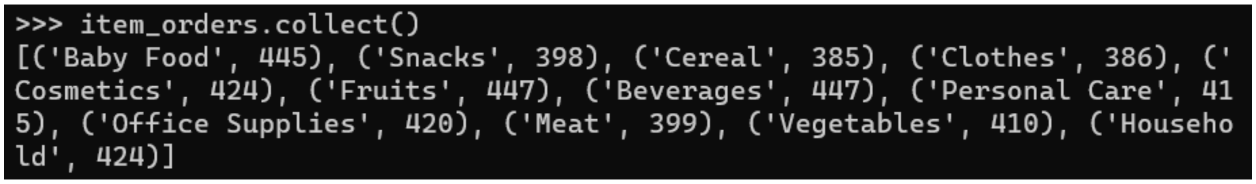


```
>>> year_wise_sales.collect()
[('2010', 3130137), ('2011', 3352394), ('2012', 3485045), ('2013', 3358584), ('2014', 3214899), ('2015', 3506548), ('2016', 3280818), ('2017', 1825066)]
>>>
```

year_wise_sales.saveAsTextFile("hdfs://localhost:9000/user/pyspark/year_wise_sales")

8.Display the number of orders for each item. (Using pyspark)

item_orders = line_rdd.map(lambda x:(x[2],1)).reduceByKey(lambda a,b:a+b)



```
>>> item_orders.collect()
[('Baby Food', 445), ('Snacks', 398), ('Cereal', 385), ('Clothes', 386), ('
Cosmetics', 424), ('Fruits', 447), ('Beverages', 447), ('Personal Care', 41
5), ('Office Supplies', 420), ('Meat', 399), ('Vegetables', 410), ('Househo
ld', 424)]
```

item_orders.saveAsTextFile("hdfs://localhost:9000/user/pyspark/item_orders")