databricks Shruthi Bhat

San Francisco Restaurant Inspection Data Analysis using SparkSQL and RDD's

General steps

- Create a case class for each data set
- Use CSV reader to read in each data file
- Convert RDD to DataFrame

Setting up input data sets

```
val baseDir = "/FileStore/tables/o7ra249x1502053895865/"
val raw_inspections = sc.textFile(s"$baseDir/inspections_plus.tsv")
val violations = sc.textFile(s"$baseDir/violations_plus.tsv")
val business = sc.textFile(s"$baseDir/businesses_plus.tsv")
```

```
baseDir: String = /FileStore/tables/o7ra249x1502053895865/
raw_inspections: org.apache.spark.rdd.RDD[String] = /FileStore/tables/o7ra249x1
502053895865//inspections_plus.tsv MapPartitionsRDD[262] at textFile at <consol
e>:36
violations: org.apache.spark.rdd.RDD[String] = /FileStore/tables/o7ra249x150205
3895865//violations_plus.tsv MapPartitionsRDD[264] at textFile at <console>:37
business: org.apache.spark.rdd.RDD[String] = /FileStore/tables/o7ra249x15020538
95865//businesses_plus.tsv MapPartitionsRDD[266] at textFile at <console>:38
```

1) What is the inspection score distribution like? (inspections_plus.csv)

Expected output - (***score, count***)

```
import org.apache.spark.sql.functions._
case class
Inspection(business_code:String,score:Int,date:String,comment:String)
val inspection=raw_inspections.map(_.split("\t"))
val validInspection=inspection.filter(l=>l(1) !=
"").map(p=>Inspection(p(0),p(1).trim.toInt,p(2),p(3)))
val InspectionDF=validInspection.toDF()
InspectionDF.groupBy("score").count().sort(desc("score")).show()
```

```
+-----+-----+
|score|count|
+-----+-----+
|  100| 3705|
|   98| 1534|
|   96| 2365|
|   94| 1751|
|   93|  374|
|   92| 1482|
|   91|  411|
|   90| 1241|
|   89|  480|
|   88|  640|
|   87|  440|
|   86|  483|
|   85|  415|
|   84|  319|
|   83|  328|
|   82|  250|
|   81|  289|
|   80|  266|
```

2) What is the risk category distribution like? (violations_plus.csv)

Expected output - (***risk category, count***)

```
import org.apache.spark.sql.functions._
case class Risk(risk:String)
val risk=violations.map(_.split("\t")).filter(l=>l(3) != "N/A")
val validRisk=risk.map(l=>Risk(l(3)))
val RiskDF=validRisk.toDF()
RiskDF.groupBy("risk").count().show()
```

```
+-------------+-----+
|         risk|count|
+-------------+-----+
|     Low Risk|24717|
|Moderate Risk|15713|
|    High Risk| 6446|
+-------------+-----+

import org.apache.spark.sql.functions._
defined class Risk
risk: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[179] at filter
at <console>:44
validRisk: org.apache.spark.rdd.RDD[Risk] = MapPartitionsRDD[180] at map at <co
nsole>:45
RiskDF: org.apache.spark.sql.DataFrame = [risk: string]
```

3) Which 20 businesses got lowest scores? (inspections_plus.csv, businesses_plus.csv)

(This should be more low score rather than lowest score)

Expected columns - (***business_id,name,address,city,postal_code,score***)

```
import org.apache.spark.sql.functions._
case class
Business(business_code:String,business_name:String,business_address:String,city
:String,postal_code:String)
val filtered_business=business.map(_.split("\t")).filter(l=>l(2)
!="").filter(l=>l(4).length ==5).map(p=>Business(p(0),p(1),p(2),p(3),p(4)))
val temp_businessDF=filtered_business.toDF()
//Replaces empty strings in the col city to San Francisco
val businessDF=temp_businessDF.na.replace("city", Map(""-> "San Francisco"))
val
grouped_inspectionDF=InspectionDF.groupBy("business_code").agg(min("score").ali
as("score"))
//val joined=businessDF.join(grouped_inspectionDF, businessDF("business_code")
=== grouped_inspectionDF("business_code"), "inner") -- this will two cols of
business_code
val joined=businessDF.join(grouped_inspectionDF,Seq("business_code"))
joined.sort("score").limit(20).show()
```

```
+-------------+--------------------+--------------------+------------+-------
----+-----+
|business_code|       business_name|    business_address|        city|postal_
```

```
code|score|
+-------------+--------------------+--------------------+-------------+-------
----+-----+
|        74522|     Dick Lee Pastry|     716 Jackson St |San Francisco|      9
4133|   42|
|        68633|     ABC Bakery Cafe|     650 Jackson St |San Francisco|      9
4133|   46|
|        18480|     Imperial Palace|  818 Washington St |San Francisco|      9
4108|   47|
|          286|  PUNJAB KABAB HOUSE|        101 EDDY St |           SF|      9
4102|   49|
|        64154|"Yummy Dim Sum & ...|    930 Stockton St |San Francisco|      9
4108|   50|
|        69962|      Hong Kee & Kim|        91 Drumm St |San Francisco|      9
4111|   51|
|         3151| New Asia Restaurant|    772 Pacific Ave |San Francisco|      9
4133|   51|
|         3115|    Yee's Restaurant|     1131 Grant Ave |San Francisco|      9
```

4) Which 20 businesses got highest scores? (inspections_plus.csv, businesses_plus.csv)

Expected columns - (***business_id,name,address,city,postal_code,score***)

```
val
max_inspectionDF=InspectionDF.groupBy("business_code").agg(max("score").alias("
score"))
val max_joined=businessDF.join(max_inspectionDF,"business_code")
max_joined.sort(desc("score")).limit(20).show()
```

```
+-------------+--------------------+--------------------+-------------+-------
----+-----+
|business_code|       business_name|    business_address|         city|postal_
code|score|
+-------------+--------------------+--------------------+-------------+-------
----+-----+
|        69528|          Toyose INC|    3814 Noriega St |           Sf|      9
4122|  100|
|         5071|          CAFE-ROSSO|  1600 HOLLOWAY Ave |           SF|      9
4132|  100|
|        34749|        Chatz Coffee|        215 02nd St |         S.F.|      9
4105|  100|
|        76060|"Edith's Food Com...|      3251 20th Ave |San Francisco|      9
4132|  100|
|         2495|STARBUCK'S COFFEE...|       2222 fillmore|           SF|      9
```

```
4115|  100|
|         5450|AT&T - (CART 3)  ...|24 WILLIE MAYS PL...|         S.F.|      9
4107|  100|
|        36744|El Castillito Taq...| 250 Golden Gate Ave|San Francisco|      9
4102|  100|
|        34989|           Cafe Muse|       785 08th Ave |           SF|      9
```

5) Among all the restaurants that got 100 score, what kind of violations did they get (if any)

(inspections_plus.csv, violations_plus.csv)

(Examine "High Risk" violation only)

Expected columns - ***(business_id, risk_category, date, description)***

Note - format the date in (***month/day/year***)

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions.{unix_timestamp, to_date,date_format}

case class
new_risk(business_code:String,date:String,violationTypeID:String,risk_category:
String,description:String)

val riskobj = violations.map(_.split("\t")).filter(l=>l(3) != "N/A").filter(l=>
(l(3)=="High Risk")).map(p=>new_risk(p(0),p(1),p(2),p(3),p(4)))
val new_riskDF=riskobj.toDF()

InspectionDF.createOrReplaceTempView("business")
val sqlDF = spark.sql("SELECT * FROM business WHERE score=100")

val joined_inspection_risk = new_riskDF.join(sqlDF,Seq("business_code","date"))
.select($"business_code",date_format(to_date(unix_timestamp($"date","yyyymmdd")
.cast("timestamp")),"MM/dd/yyyy").alias("date"),$"risk_category",
  $"description").show()
```

```
+-------------+----------+-------------+--------------------+
|business_code|      date|risk_category|         description|
+-------------+----------+-------------+--------------------+
|        18825|01/21/2014|    High Risk|No hot water or r...|
|         1896|01/11/2014|    High Risk|Improper reheatin...|
|           17|01/23/2012|    High Risk|High risk food ho...|
|         5874|01/27/2014|    High Risk|Improper reheatin...|
|         3482|01/10/2015|    High Risk|Unclean or unsani...|
|        36744|01/26/2013|    High Risk|Improper cooling ...|
+-------------+----------+-------------+--------------------+

import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions.{unix_timestamp, to_date, date_format}
defined class new_risk
riskobj: org.apache.spark.rdd.RDD[new_risk] = MapPartitionsRDD[222] at map at <
console>:65
new_riskDF: org.apache.spark.sql.DataFrame = [business_code: string, date: stri
ng ... 3 more fields]
sqlDF: org.apache.spark.sql.DataFrame = [business_code: string, score: int ...
2 more fields]
joined_inspection_risk: Unit = ()
```

6) Average inspection score by zip code

Expected columns - (***zip, average score with only two digits after decimal***)

```
//Find the avg for each restuarant
val
avg_inspectionDF=InspectionDF.groupBy("business_code").agg(avg("score").alias("
score"))
val joined_business_inspection=
businessDF.join(avg_inspectionDF,"business_code").groupBy("postal_code").agg(ro
und(avg($"score"),2).alias("score")).show(40)
```

```
+-----------+-----+
|postal_code|score|
+-----------+-----+
|      94102| 90.9|
|      94140| 96.5|
|      94107|94.76|
|      94104|93.57|
|      94131|93.21|
|      94014| 90.0|
|      94143| 89.5|
|      94609| 87.0|
|      94112|92.82|
|      94545|100.0|
|      92672|91.33|
|      94513|92.33|
|      94103|90.58|
|      94130|97.75|
|      94118|92.64|
|      94117| 91.6|
|      94129| 83.0|
```

7) Compute the proportion of all businesses in each neighborhood that have incurred at least one of the violations

- "High risk vermin infestation"
- "Moderate risk vermin infestation"
- "Sewage or wastewater contamination”
- "Improper food labeling or menu misrepresentation"
- "Contaminated or adulterated food”
- "Reservice of previously served foods"
- "Expected output: zip code, percentage"

This question is asking for each neighborhood, what is the proportion of businesses that have incurred at least one of the above nasty violations

Note: use UDF to determine which violations match with one of the above extremely bad violations

Expected columns - (***zip code, total violation count, extreme violation count, proportion with only two digits after decimal***)

```
import spark.implicits._
import org.apache.spark.sql.functions._
//create a udf
def extremeViolation(desc:String) : Int = {

  val violationList=List("High risk vermin infestation","Moderate risk vermin
infestation","Sewage or wastewater contamination","Improper food labeling or
menu misrepresentation","Contaminated or adulterated food","Reservice of
previously served foods")
  if (violationList.contains(desc)) return 1
  else return 0
}

//Initialize the udf
val ExtremeViolationUDF = udf(extremeViolation(_:String))

//Get the required fields from the tsv file
case class new_business(business_code:String,zip:String)
val new_value=business.map(_.split("\t")).filter(l=>l(4).length
==5).map(p=>new_business(p(0),p(4))).toDF()
case class new_violation(business_code:String,violations:String)
val
temp_value=violations.map(_.split("\t")).filter(l=>l.length==5).filter(l=>l(4)
!= "").map(p=>new_violation(p(0),p(4))).distinct().toDF()



val joined_table=new_value.join(temp_value,Seq("business_code"))
val res1=joined_table.groupBy($"zip").agg(count("*").alias("total violation
count"))
val
res2=joined_table.select($"zip",ExtremeViolationUDF($"violations").as("vc")).gr
oupBy($"zip").agg(round(sum($"vc"),2).alias("extreme violation count"))
val final_res=res1.join(res2,Seq("zip")).select($"zip",$"total violation
count",$"extreme violation count",round((($"extreme violation count"/$"total
violation count")*100),2) as "proportion in percentage")

final_res.sort(desc("proportion in percentage")).show(100)
```

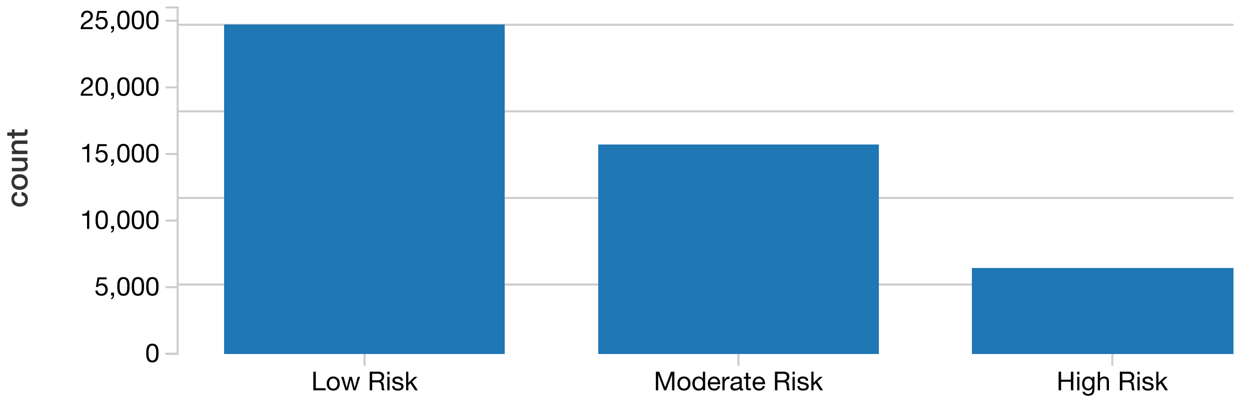
```
+-----+---------------------+-----------------------+-----------------------+
|  zip|total violation count|extreme violation count|proportion in percentage|
+-----+---------------------+-----------------------+-----------------------+
|94129|                    6|                      1|                  16.67|
|94115|                 1357|                    135|                   9.95|
|94109|                 2582|                    236|                   9.14|
|94143|                   22|                      2|                   9.09|
|94133|                 3291|                    299|                   9.09|
|94122|                 1948|                    171|                       8.78|
|94110|                 3142|                    255|                       8.12|
|94134|                  457|                     37|                        8.1|
|94116|                  602|                     47|                       7.81|
|94158|                   13|                      1|                       7.69|
|94114|                 1283|                     97|                       7.56|
|94108|                 1515|                    114|                       7.52|
|94117|                 1027|                     76|                        7.4|
|94102|                 2438|                    172|                       7.05|
|94112|                 1079|                     75|                       6.95|
|94123|                 1284|                     75|                       5.84|
|94124|                  730|                     41|                       5.62|
|94104|                  565|                     31|                       5.49|
```

8) Are SF restaurants clean? Justify your answer

(***Make to sure backup your answer with data - don't just state your opinion***)

(Yes/No) and why

I feel SF restuarants are not extremely clean.If you see the violation for high and moderate risk, it is almost equal to 50% of the entire inspection dataset


count
Low Risk
Moderate Risk
High Risk