Assignment 19.1

Downloaded the Dataset an uploaded in the location **/home/acadgild/sumona**

1. What are the total number of gold medal winners every year

First we will create a RDD which will read the dataset from the local file system

val baseRDD = sc.textFile("/home/acadgild/sumona/Sports_data.txt")

```
scala> val baseRDD = sc.textFile("/home/acadgild/sumona/Sports_data.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/sumona/Sports_data.txt MapPartitionsRDD[1] at textFile at <console>:26

scala> baseRDD.collect().foreach(println)
firstname,lastname,sports,medal_type,age,year,country
lisa,cudrow,javellin,gold,34,2015,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2016,USA
usha,pt,running,silver,30,2016,IND
serena,williams,running,gold,31,2014,FRA
roger,federer,tennis,silver,32,2016,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2016,CHN
lisa,cudrow,javellin,gold,34,2017,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2017,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2014,USA
mathew,louis,javellin,gold,34,2014,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2014,CHN
jenifer,cox,swimming,silver,32,2017,IND
fernando,johnson,swimming,silver,32,2017,CHN

scala>
```

We will define schema's since the input file is a text file.

val schemaString =
"firstname:string,lastname:string,sports:string,medal:string,age:int,year:int,country:string"

val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if(x.split(":")(1).equals("string")) StringType else IntegerType, true)))

```
scala> val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string"
schemaString: String = firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string

scala>

scala> val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if(x.split(":")(1).equals("string")) StringType else IntegerType, tr
ue)))
schema: org.apache.spark.sql.types.StructType = StructType(StructField(firstname,StringType,true), StructField(lastname,StringType,true), StructField(sports,Stri
ngType,true), StructField(medal,StringType,true), StructField(age,IntegerType,true), StructField(year,IntegerType,true), StructField(country,StringType,true))

scala>
```

Then, we will split the input file and extract the rows

val rowRDD = baseRDD.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt,r(5).toInt, r(6)))

```
scala> val rowRDD = baseRDD.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt,r(5).toInt, r(6)))
rowRDD: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[11] at map at <console>:28

scala>
```

```
scala> rowRDD.collect().foreach(println)
[firstname,lastname,sports,medal_type,age,year,country]
[lisa,cudrow,javellin,gold,34,2015,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2016,USA]
[usha,pt,running,silver,30,2016,IND]
[serena,williams,running,gold,31,2014,FRA]
[roger,federer,tennis,silver,32,2016,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2016,CHN]
[lisa,cudrow,javellin,gold,34,2017,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2017,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]
[lisa,cudrow,javellin,gold,34,2014,USA]
[mathew,louis,javellin,gold,34,2014,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2014,CHN]
[jenifer,cox,swimming,silver,32,2017,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]

scala>
```

Now we will create a dataframe by passing the RDD which reads the rowRDD and schema

val SportsDataDF = spark.createDataFrame(rowRDD, schema)

```
scala> val SportsDataDF = spark.createDataFrame(rowRDD, schema)
18/01/08 12:13:01 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
SportsDataDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 5 more fields]

scala> SportsDataDF.printSchema()
root
 |-- firstname: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- sports: string (nullable = true)
 |-- medal: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- country: string (nullable = true)


scala>
```

Here, we will now create a table from the dataframe and the execute the SQL query to get the total number of gold medal winners every year

SportsDataDF.createOrReplaceTempView("Sports_Data")

val result1DF = spark.sql("SELECT year,COUNT(*) FROM Sports_Data WHERE medal = 'gold'GROUP BY year")

result1DF.show()

```
scala> SportsDataDF.createOrReplaceTempView("Sports_Data")

scala>

scala> val result1DF = spark.sql("SELECT year,COUNT(*) FROM Sports_Data WHERE medal = 'gold'GROUP BY year")
result1DF: org.apache.spark.sql.DataFrame = [year: string, count(1): bigint]

scala>

scala> result1DF.show()
+----+--------+
|year|count(1)|
+----+--------+
|2016|       2|
|2017|       1|
|2014|       3|
|2015|       3|
+----+--------+


scala>
```

2. How many silver medals have been won by USA in each sport

val result2DF = spark.sql("SELECT sports,COUNT(*) FROM Sports_Data WHERE medal = 'silver'and country ='USA' GROUP BY sports")

```
scala> val result2DF = spark.sql("SELECT sports,COUNT(*) FROM Sports_Data WHERE medal = 'silver'and country ='USA' GROUP BY sports")
result2DF: org.apache.spark.sql.DataFrame = [sports: string, count(1): bigint]

scala> result2DF.show()
+--------+--------+
|  sports|count(1)|
+--------+--------+
|swimming|       3|
+--------+--------+


scala>
```