Task 1
Using spark-sql, Find:

```
import org.apache.spark.sql.functions.udf

import org.apache.spark.sql.catalyst.encoders.ExpressionEncoder

import org.apache.spark.sql.Encoder

import org.apache.spark.sql.{Row, SparkSession}

import org.apache.spark.sql.types.{DoubleType, StringType, StructField, StructType}

import spark.implicits._


case class Sports(firstname: String, lastname: String, sports: String, medal_type: String, age: Int, year: Int, country: String)


val sportsDF = sc.textFile("/Session21/Sports_dataNC.txt")


val sportsDFmap =
sportsDF.map(_.split(",")).toDF("firstname","lastname","sports","medal_type","age","year","co
untry")


val sportsDFmap = sportsDF.map(_.split(","))

val SportsData = sportsDFmap.map(attributes => Sports(attributes(0), attributes(1),
attributes(2), attributes(3), attributes(4).trim.toInt, attributes(5).trim.toInt, attributes(6).trim))

val SportsSQL = SportsData.toDF()

SportsSQL.show()
```

```
scala> val sportsDF = sc.textFile("/Session21/Sports_dataNC.txt")
sportsDF: org.apache.spark.rdd.RDD[String] = /Session21/Sports_dataNC.txt MapPar
titionsRDD[21] at textFile at <console>:48

scala>

scala>

scala> val sportsDFmap = sportsDF.map(_.split(",")).toDF("firstname","lastname",
"sports","medal_type","age","year","country")
java.lang.IllegalArgumentException: requirement failed: The number of columns do
esn't match.
Old column names (1): value
New column names (7): firstname, lastname, sports, medal_type, age, year, countr
y
  at scala.Predef$.require(Predef.scala:224)
  at org.apache.spark.sql.Dataset.toDF(Dataset.scala:393)
  at org.apache.spark.sql.DatasetHolder.toDF(DatasetHolder.scala:44)
  ... 54 elided

scala>

scala>

scala> val sportsDFmap = sportsDF.map(_.split(","))
sportsDFmap: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[23] at m
ap at <console>:50

scala>

scala> val SportsData = sportsDFmap.map(attributes => Sports(attributes(0), attr
ibutes(1), attributes(2), attributes(3), attributes(4).trim.toInt, attributes(5)
.trim.toInt, attributes(6).trim))
SportsData: org.apache.spark.rdd.RDD[Sports] = MapPartitionsRDD[24] at map at <c
onsole>:54

scala>

scala> val SportsSQL = SportsData.toDF()
SportsSQL: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string
 ... 5 more fields]

scala>

scala> SportsSQL.show()
+---------+--------+--------+----------+---+----+-------+
|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|
+---------+--------+--------+----------+---+----+-------+
only showing top 20 rows


scala>
```

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

```
scala> val filterGold = getMedals.filter($"medal_type" === "gold")
filterGold: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [year: int,
 medal_type: string]

scala> filterGold.show()
+----+----------+
|year|medal_type|
+----+----------+
|2015|      gold|
|2015|      gold|
|2014|      gold|
|2017|      gold|
|2015|      gold|
|2016|      gold|
|2014|      gold|
|2014|      gold|
|2016|      gold|
+----+----------+


scala> val ResultGold = filterGold.groupBy("year").count()
ResultGold: org.apache.spark.sql.DataFrame = [year: int, count: bigint]

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


scala>
```

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

```
+---------+--------+--------+----------+---+----+-------+
|     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|
+---------+--------+--------+----------+---+----+-------+
only showing top 20 rows


scala> val USASilver = SportsSQL.Select($"sports", $"medal_type").filter($"medal
_type" === "silver")
<console>:58: error: value Select is not a member of org.apache.spark.sql.DataFr
ame
       val USASilver = SportsSQL.Select($"sports", $"medal_type").filter($"medal
_type" === "silver")
                           ^

scala> val USASilver = SportsSQL.select($"sports", $"medal_type").filter($"medal
_type" === "silver")
USASilver: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [sports: str
ing, medal_type: string]

scala> USASilver.show()
+--------+----------+
|  sports|medal_type|
+--------+----------+
|swimming|    silver|
| running|    silver|
|  tennis|    silver|
|swimming|    silver|
|swimming|    silver|
|swimming|    silver|
| running|    silver|
|  tennis|    silver|
|swimming|    silver|
|swimming|    silver|
|swimming|    silver|
| running|    silver|
|  tennis|    silver|
|swimming|    silver|
|swimming|    silver|
+--------+----------+


scala> val USASilverCnt = USASilver.groupBy("sports").count()
USASilverCnt: org.apache.spark.sql.DataFrame = [sports: string, count: bigint]

scala> USASilverCnt.show()
+--------+-----+
|  sports|count|
+--------+-----+
| running|    3|
|swimming|    9|
|  tennis|    3|
+--------+-----+


scala>
```

Task 2
Using udfs on dataframe

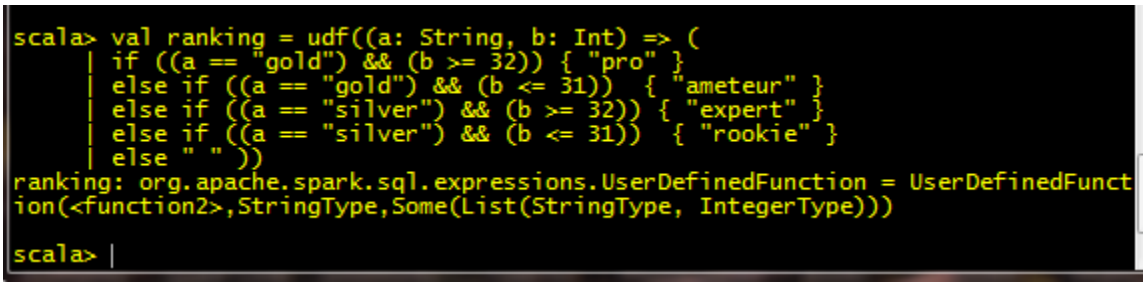
1. Change firstname, lastname columns into

Mr.first_two_letters_of_firstname<space>lastname

for example - michael, phelps becomes Mr.mi phelps

```
scala> sportsNames.show()
+---------+--------+
|firstname|lastname|
+---------+--------+
|     lisa|  cudrow|
|   mathew|   louis|
|  michael|  phelps|
|     usha|      pt|
|   serena|williams|
|    roger| federer|
|  jenifer|     cox|
| fernando| johnson|
|     lisa|  cudrow|
|   mathew|   louis|
|  michael|  phelps|
|     usha|      pt|
|   serena|williams|
|    roger| federer|
|  jenifer|     cox|
| fernando| johnson|
|     lisa|  cudrow|
|   mathew|   louis|
|  michael|  phelps|
|     usha|      pt|
+---------+--------+
only showing top 20 rows


scala> val convertName = udf((a: String, b: String) => String = "Mr."+a.take(2)+
 " "+b)
<console>:47: error: object java.lang.String is not a value
       val convertName = udf((a: String, b: String) => String = "Mr."+a.take(2)+
 " "+b)
                                                           ^

scala> val convertName = udf((a: String, b: String) => "Mr."+a.take(2)+ " "+b)
convertName: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedF
unction(<function2>,StringType,Some(List(StringType, StringType)))

scala> val namesConverted = sportsNames.select(convertName($"firstname",$"lastna
me"))
namesConverted: org.apache.spark.sql.DataFrame = [UDF(firstname, lastname): stri
ng]

scala> namesConverted.show()
+-----------------------+
|UDF(firstname, lastname)|
+-----------------------+
|             Mr.li cudrow|
|              Mr.ma louis|
|             Mr.mi phelps|
|                 Mr.us pt|
|           Mr.se williams|
|             Mr.ro federer|
|                 Mr.je cox|
|             Mr.fe johnson|
|             Mr.li cudrow|
|              Mr.ma louis|
|             Mr.mi phelps|
|                 Mr.us pt|
|           Mr.se williams|
|             Mr.ro federer|
|                 Mr.je cox|
|             Mr.fe johnson|
|             Mr.li cudrow|
|              Mr.ma louis|
|             Mr.mi phelps|
|                 Mr.us pt|
+-----------------------+
only showing top 20 rows


scala>
```

2. Add a new column called ranking using udfs on dataframe, where :
gold medalist, with age >= 32 are ranked as pro
gold medalists, with age <= 31 are ranked amateur
silver medalist, with age >= 32 are ranked as expert
silver medalists, with age <= 31 are ranked rookie



```
scala> SportsSQL.show()
+---------+--------+--------+----------+---+----+-------+
|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|
+---------+--------+--------+----------+---+----+-------+
only showing top 20 rows


scala> val showRanking = SportsSQL.withColumn("ranking", ranking($"medal_type",$
"age"))
showRanking: org.apache.spark.sql.DataFrame = [firstname: string, lastname: stri
ng ... 6 more fields]

scala> showRanking.show()
+---------+--------+--------+----------+---+----+-------+-------+
|firstname|lastname|  sports|medal_type|age|year|country|ranking|
+---------+--------+--------+----------+---+----+-------+-------+
|     lisa|  cudrow|javellin|      gold| 34|2015|    USA|    pro|
|   mathew|   louis|javellin|      gold| 34|2015|    RUS|    pro|
|  michael|  phelps|swimming|    silver| 32|2016|    USA| expert|
|     usha|      pt| running|    silver| 30|2016|    IND| rookie|
|   serena|williams| running|      gold| 31|2014|    FRA|ameteur|
|    roger| federer|  tennis|    silver| 32|2016|    CHN| expert|
|  jenifer|     cox|swimming|    silver| 32|2014|    IND| expert|
| fernando| johnson|swimming|    silver| 32|2016|    CHN| expert|
|     lisa|  cudrow|javellin|      gold| 34|2017|    USA|    pro|
|   mathew|   louis|javellin|      gold| 34|2015|    RUS|    pro|
|  michael|  phelps|swimming|    silver| 32|2017|    USA| expert|
|     usha|      pt| running|    silver| 30|2014|    IND| rookie|
|   serena|williams| running|      gold| 31|2016|    FRA|ameteur|
|    roger| federer|  tennis|    silver| 32|2017|    CHN| expert|
|  jenifer|     cox|swimming|    silver| 32|2014|    IND| expert|
| fernando| johnson|swimming|    silver| 32|2017|    CHN| expert|
|     lisa|  cudrow|javellin|      gold| 34|2014|    USA|    pro|
|   mathew|   louis|javellin|      gold| 34|2014|    RUS|    pro|
|  michael|  phelps|swimming|    silver| 32|2017|    USA| expert|
|     usha|      pt| running|    silver| 30|2014|    IND| rookie|
+---------+--------+--------+----------+---+----+-------+-------+
only showing top 20 rows


scala>
```