

ASSIGNMENT 21.1

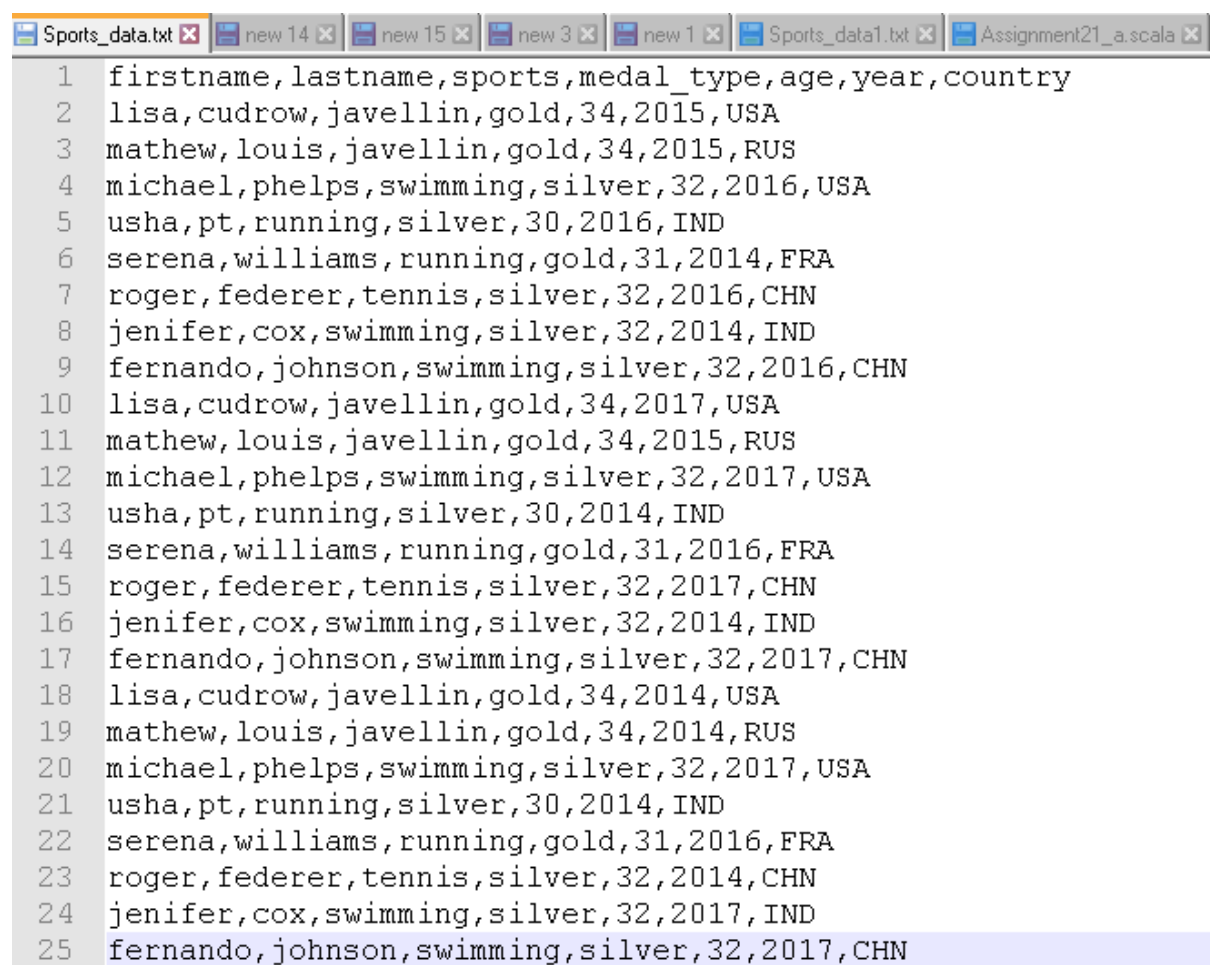
I have create this assignment in **IntelliJIDEA** application for scala.

To solve the all problems, I create two scala file with objects **Assignment21_Task1** and **Assignment21_Task2**.

Description of all codes explained in the code file.

Below screen shots show the input dataset and output obtained by the code for each problem in IntelliJIDEA application.

Below screen, shot shows the input data-set Sports-data.txt



```
1  firstname,lastname,sports,medal_type,age,year,country
2  lisa,cudrow,javellin,gold,34,2015,USA
3  mathew,louis,javellin,gold,34,2015,RUS
4  michael,phelps,swimming,silver,32,2016,USA
5  usha,pt,running,silver,30,2016,IND
6  serena,williams,running,gold,31,2014,FRA
7  roger,federer,tennis,silver,32,2016,CHN
8  jenifer,cox,swimming,silver,32,2014,IND
9  fernando,johnson,swimming,silver,32,2016,CHN
10 lisa,cudrow,javellin,gold,34,2017,USA
11 mathew,louis,javellin,gold,34,2015,RUS
12 michael,phelps,swimming,silver,32,2017,USA
13 usha,pt,running,silver,30,2014,IND
14 serena,williams,running,gold,31,2016,FRA
15 roger,federer,tennis,silver,32,2017,CHN
16 jenifer,cox,swimming,silver,32,2014,IND
17 fernando,johnson,swimming,silver,32,2017,CHN
18 lisa,cudrow,javellin,gold,34,2014,USA
19 mathew,louis,javellin,gold,34,2014,RUS
20 michael,phelps,swimming,silver,32,2017,USA
21 usha,pt,running,silver,30,2014,IND
22 serena,williams,running,gold,31,2016,FRA
23 roger,federer,tennis,silver,32,2014,CHN
24 jenifer,cox,swimming,silver,32,2017,IND
25 fernando,johnson,swimming,silver,32,2017,CHN
```

```

val sports_data_with_header =
spark.sparkContext.textFile("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\AcadgildSessions\\Se
ssion21_Spark_SQL2\\Sports_data.txt")

val header = sports_data_with_header.first()

val sports_data = sports_data_with_header.filter(row => row != header)

val sports_data_df = sports_data.map(_._split(",")).map(x => SPORTS(x(0), x(1), x(2), x(3),
x(4), x(5), x(6))).toDF

sports_data_df.show()

```

Below screen shot shows input data-set sports_data_df without header line

```

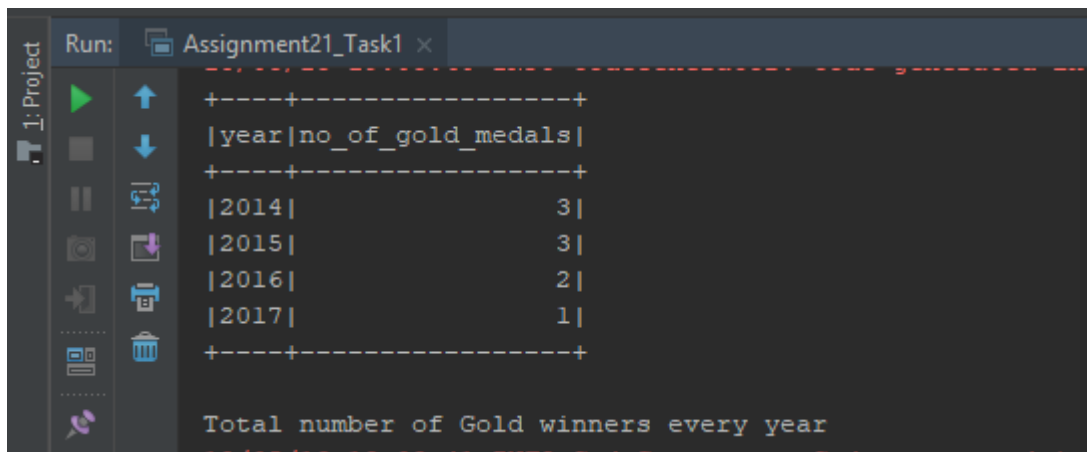
+-----+-----+-----+-----+-----+-----+
|firstname|lastname| sports|medal_type|age|year|country|
+-----+-----+-----+-----+-----+
| lisa| cudrow|javellin| gold| 34|2015| USA|
| mathew| louis|javellin| gold| 34|2015| RUS|
| michael| phelps|swimming| silver| 32|2016| USA|
| usha| pt| running| silver| 30|2016| 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

```

Using spark-sql, Find:

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

```
val no_of_gold_winners = spark.sql("SELECT year, count(*) AS gold_medals FROM sports  
WHERE medal_type='gold' GROUP BY year ORDER BY year").show()
```



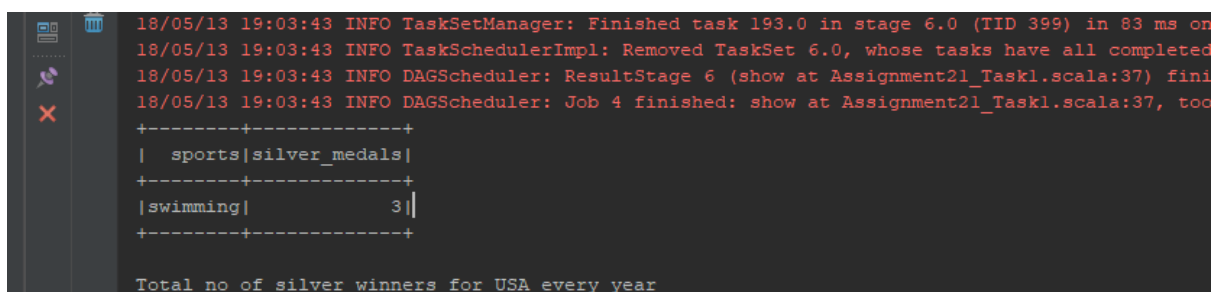
The screenshot shows a terminal window with a Spark SQL query result. The query is: `SELECT year, count(*) AS gold_medals FROM sports WHERE medal_type='gold' GROUP BY year ORDER BY year`. The result is displayed in a table format with columns `year` and `no_of_gold_medals`. The data shows 3 gold medals in 2014 and 2015, 2 in 2016, and 1 in 2017. Below the table, a summary line states: "Total number of Gold winners every year".

year	no_of_gold_medals
2014	3
2015	3
2016	2
2017	1

Total number of Gold winners every year

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

```
val no_of_silver_winners_ = spark.sql("SELECT sports, count(*) AS silver_medals FROM  
sports WHERE country='USA' and medal_type='silver' GROUP BY sports ORDER BY  
sports").show()
```



The screenshot shows a terminal window with a Spark SQL query result. The query is: `SELECT sports, count(*) AS silver_medals FROM sports WHERE country='USA' and medal_type='silver' GROUP BY sports ORDER BY sports`. The result is displayed in a table format with columns `sports` and `silver_medals`. The data shows 3 silver medals for swimming. Below the table, a summary line states: "Total no of silver winners for USA every year".

sports	silver_medals
swimming	3

Total no of silver winners for USA every year

Task 2

Using udfs on dataframe

a. Change firstname, lastname columns into

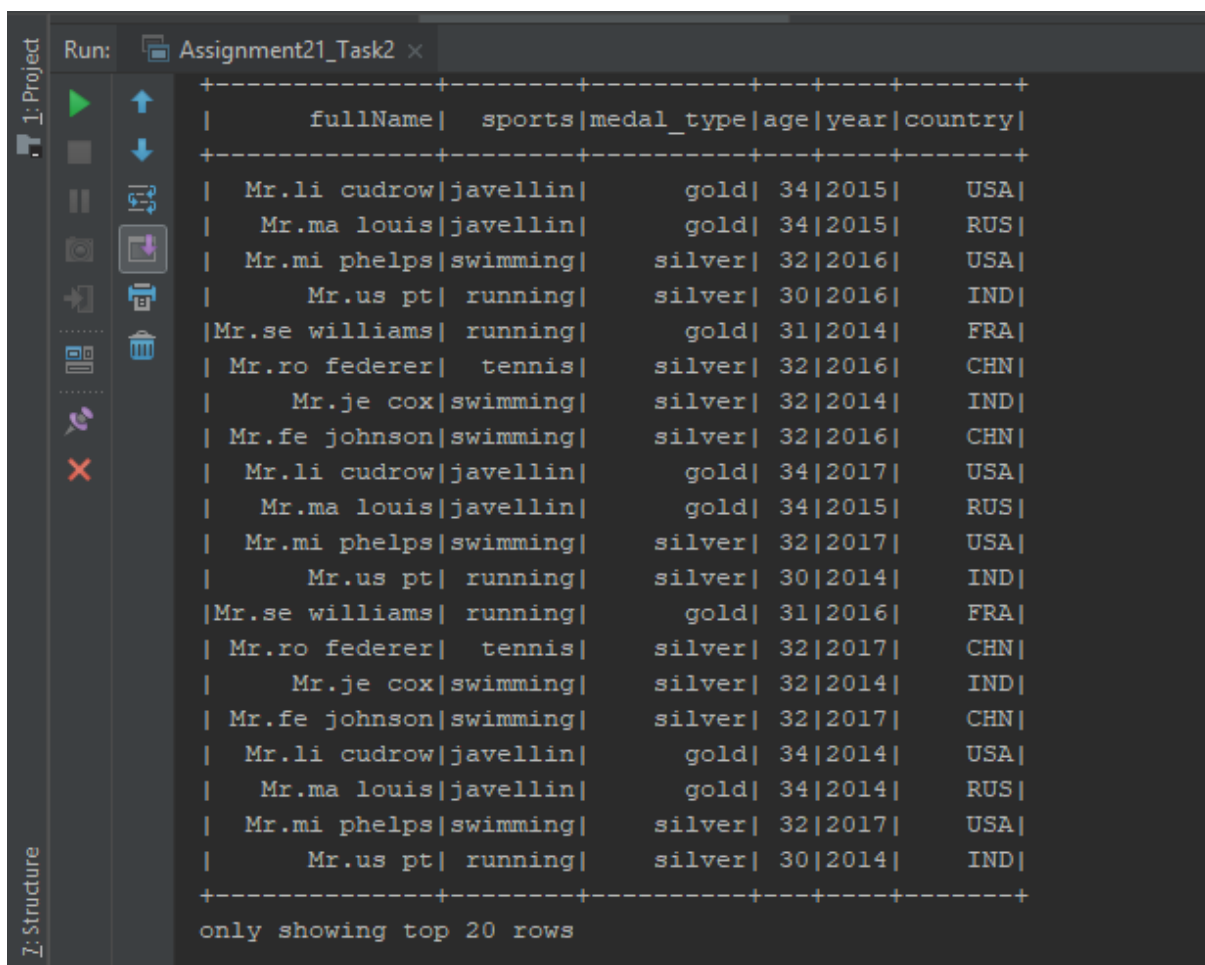
Mr.first_two_letters_of_firstname<space>lastname

for example - michael, phelps becomes Mr.mi phelps

```
val first_and_last_name_concat = udf((first_name: String, last_name: String) =>
"Mr.".concat(first_name.substring(0, 2)).concat(" ").concat(last_name))
```

```
val new_sports_data_sports_df = sports_data_df.withColumn("fullName",
first_and_last_name_concat(sports_data_df("firstname"), sports_data_df("lastname")))
```

```
new_sports_data_sports_df.select("fullName","sports","medal_type","age","year","country")
.show()
```



fullName	sports	medal_type	age	year	country
Mr.li cudrow	javellin	gold	34	2015	USA
Mr.ma louis	javellin	gold	34	2015	RUS
Mr.mi phelps	swimming	silver	32	2016	USA
Mr.us pt	running	silver	30	2016	IND
Mr.se williams	running	gold	31	2014	FRA
Mr.ro federer	tennis	silver	32	2016	CHN
Mr.je cox	swimming	silver	32	2014	IND
Mr.fe johnson	swimming	silver	32	2016	CHN
Mr.li cudrow	javellin	gold	34	2017	USA
Mr.ma louis	javellin	gold	34	2015	RUS
Mr.mi phelps	swimming	silver	32	2017	USA
Mr.us pt	running	silver	30	2014	IND
Mr.se williams	running	gold	31	2016	FRA
Mr.ro federer	tennis	silver	32	2017	CHN
Mr.je cox	swimming	silver	32	2014	IND
Mr.fe johnson	swimming	silver	32	2017	CHN
Mr.li cudrow	javellin	gold	34	2014	USA
Mr.ma louis	javellin	gold	34	2014	RUS
Mr.mi phelps	swimming	silver	32	2017	USA
Mr.us pt	running	silver	30	2014	IND

only showing top 20 rows

b. Add a new column called ranking using udfs on dataframe,

where :

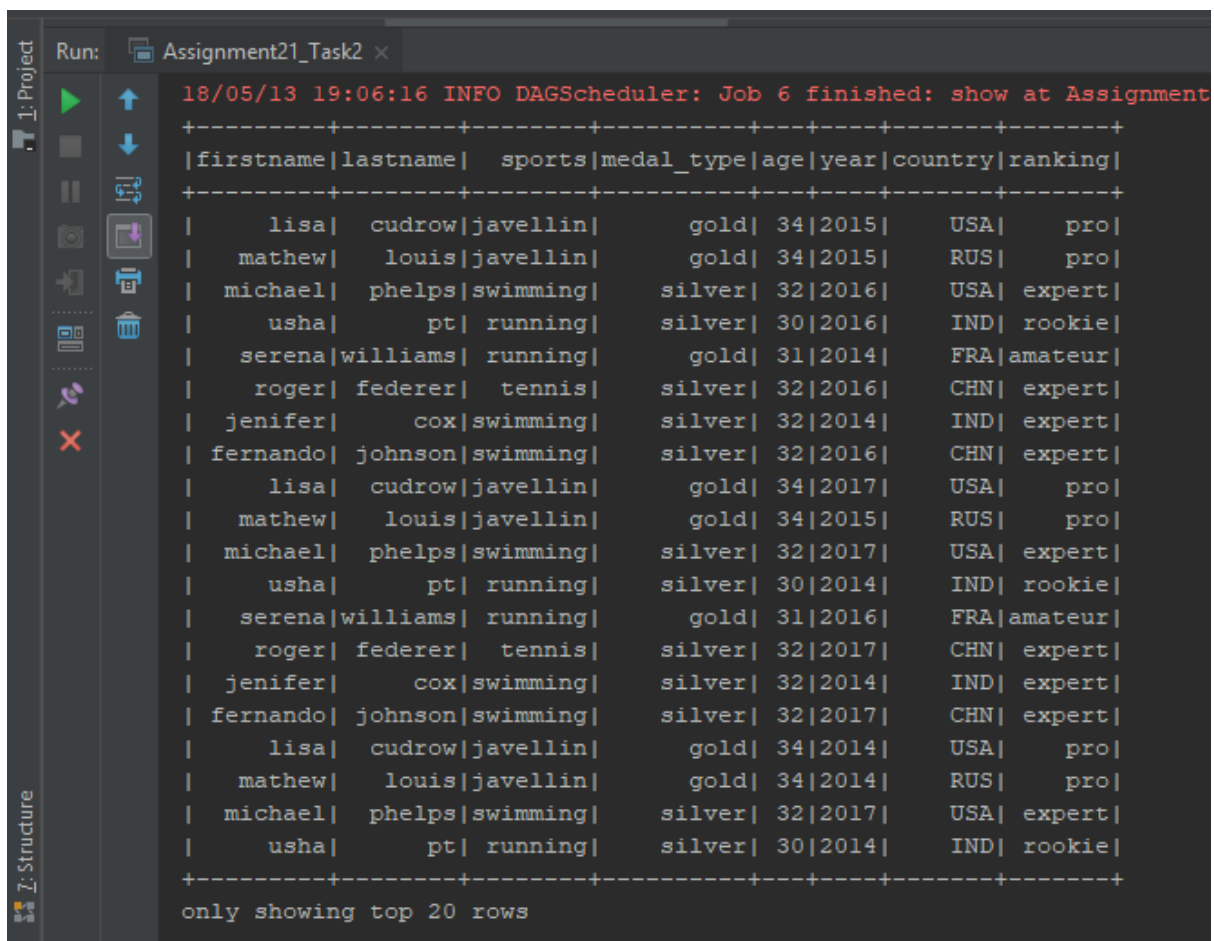
gold medalist, with age ≥ 32 are ranked as pro

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

```
val Rank = udf((medal_type: String, age: Int) => {  
    if (medal_type == "gold" && age  $\geq$  32) "pro"  
    else if (medal_type == "gold" && age  $\leq$  31) "amateur"  
    else if (medal_type == "silver" && age  $\geq$  32) "expert"  
    else if (medal_type == "silver" && age  $\leq$  31) "rookie" else "no-level" })  
sports_data_df.withColumn("ranking", Rank(sports_data_df("medal_type"),  
sports_data_df("age"))).show()
```



Run: Assignment21_Task2 x

18/05/13 19:06:16 INFO DAGScheduler: Job 6 finished: show at Assignment

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	amateur
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	amateur
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