STA323 Assignment 2 report

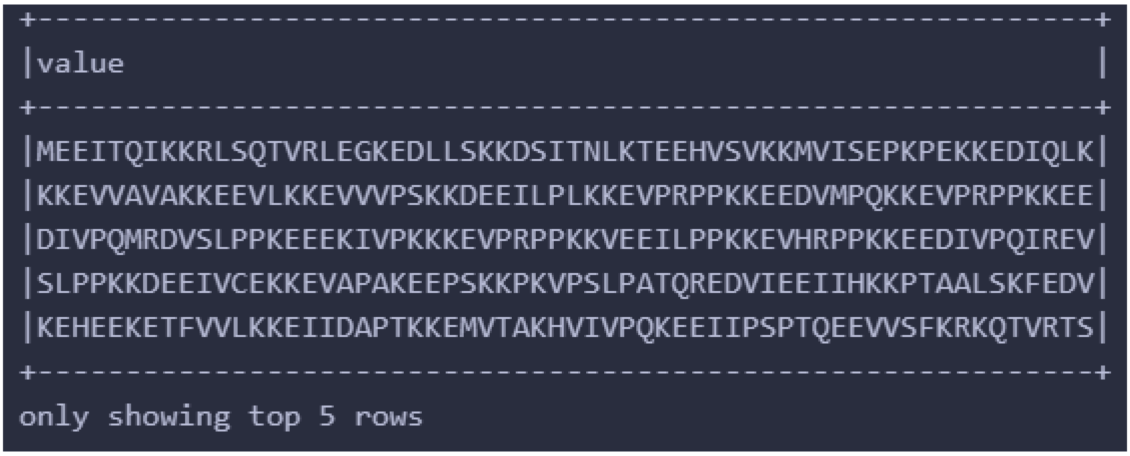
SID: 12110821
Name: ZHANG Chi

Solution for Q1

(1)

As the header of each seqence is started with `>` , we can drop it by `filter` after reading the file by `spark.read.text` .

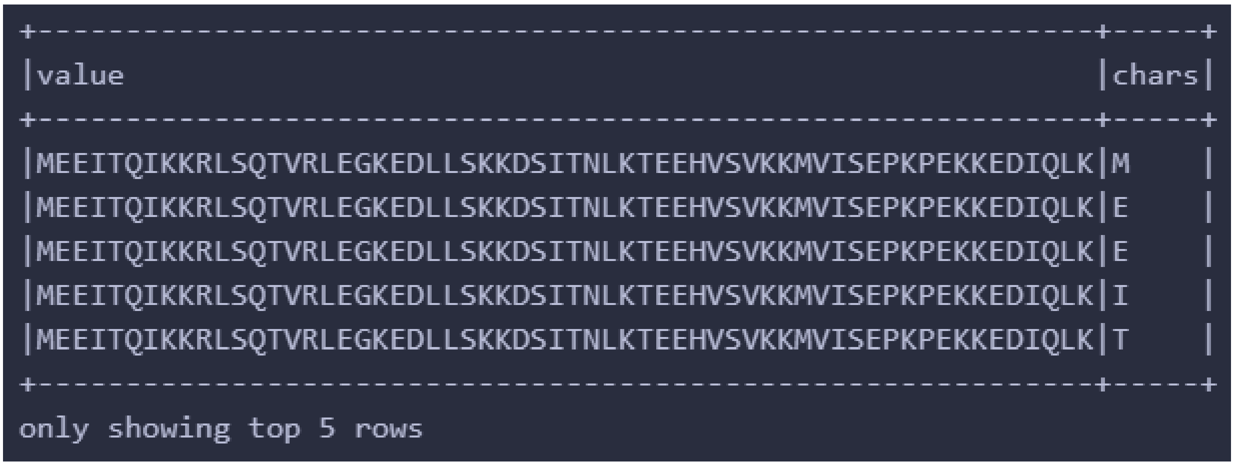
```
df1_1 = spark.read.text("data/Q1_data/protein.fasta")
df1_1 = df1_1.filter(~col("value").contains(">"))
```



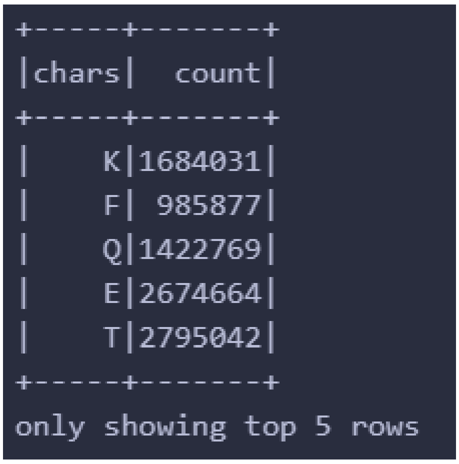
```
+------------------------------------------------------------+
|value                                                       |
+------------------------------------------------------------+
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|
|KKEVVAVAKKEEVLKKEVVVPSKKDEEILPLKKEVPRPPKKEEDVMPQKKEVPRPPKKEE|
|DIVPQMRDVSLPPKEEEKIVPKKKEVPRPPKKVEEILPPKKEVHRPPKKEEDIVPQIREV|
|SLPPKKDEEIVCEKKEVAPAKEEPSKKPKVPSLPATQREDVIEEIIHKKPTAALSKFEDV|
|KEHEEKETFVVLKKEIIDAPTKKEMVTAKHVIVPQKEEIIPSPTQEEVVSFKRKQTVRTS|
+------------------------------------------------------------+
only showing top 5 rows
```

Then we can split each line by `split` and `explode` them to make every char (amino acid) in one line in the column `chars` . Finally, we can use `groupBy` and `count` to get the frequency of each word.

```
df1_1_withchars = df1_1.withColumn("chars", explode(split(col("value"), "")))
df1_1_withchars.groupBy("chars").count().show(5)
```



```
+------------------------------------------------------------+-----+
|value                                                       |chars|
+------------------------------------------------------------+-----+
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|M    |
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|E    |
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|E    |
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|I    |
|MEEITQIKKRLSQTVRLEGKEDLLSKKDSITNLKTEEHVSVKKMVISEPKPEKKEDIQLK|T    |
+------------------------------------------------------------+-----+
only showing top 5 rows
```



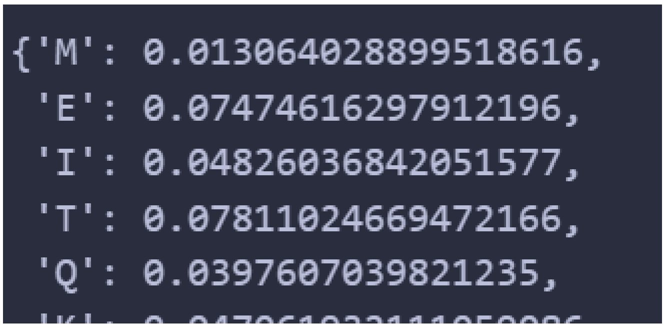
```
+-----+-------+
|chars|  count|
+-----+-------+
|    K|1684031|
|    F| 985877|
|    Q|1422769|
|    E|2674664|
|    T|2795042|
+-----+-------+
only showing top 5 rows
```

(2)

Using RDD api, we can read the file and drop the header by `filter` and `map` the lines to characters. Then we can use `flatMap` to make every char in one line and use `countByValue` to get the frequency of each word.

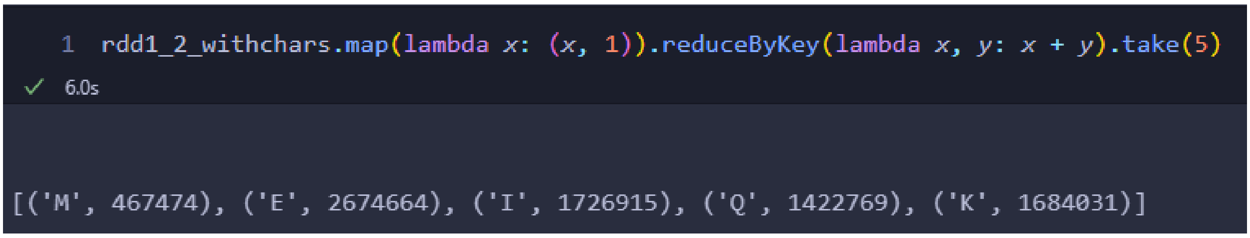
```
rdd1_2 = spark.sparkContext.textFile("data/Q1_data/protein.fasta")
rdd1_2 = rdd1_2.filter(lambda x: ">" not in x)
rdd1_2_withchars = rdd1_2.flatMap(lambda x: list(x))
total_count = rdd1_2_withchars.count()

frequencies = {k: v / total_count for k, v in counts.items()}
frequencies
```



```
{'M': 0.013064028899518616,
 'E': 0.07474616297912196,
 'I': 0.04826036842051577,
 'T': 0.0781102466947216б,
 'Q': 0.039760703982123 5,
```

Besides, we can also use `reduceByKey` to get the count of each word, which could be more friendly to a low memory driver.



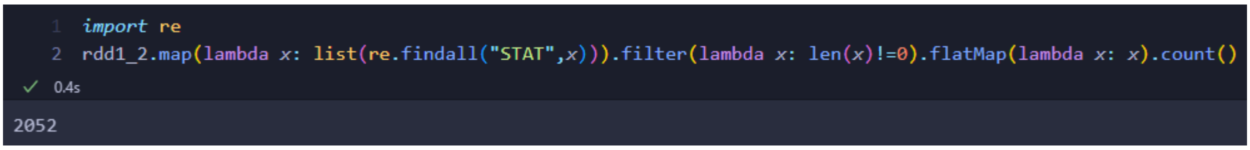
```
rdd1_2_withchars.map(lambda x: (x, 1)).reduceByKey(lambda x, y: x + y).take(5)
```

✓ 6.0s

```
[('M', 467474), ('E', 2674664), ('I', 1726915), ('Q', 1422769), ('K', 1684031)]
```

(3)

To find the a specific sequence motif "STAT" (omitting the line break), we can use `re.findall` in `re` module to find all the matches in the every element of the rdd object `rdd1_2`. After that, we can use `filter` to drop the elements without any match and use `flatMap` to make every match in one line. Finally, we can use `count` to get the number of the matches.



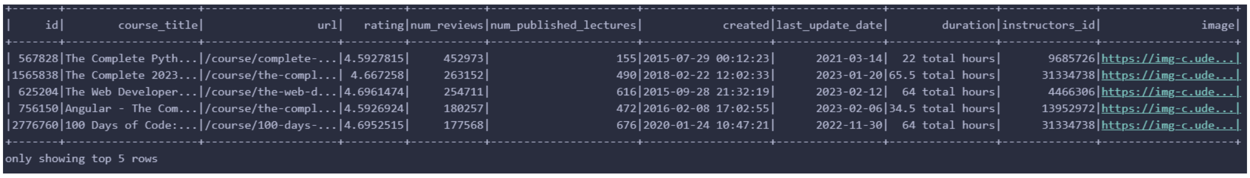
```
import re
rdd1_2.map(lambda x: list(re.findall("STAT",x))).filter(lambda x: len(x)!=0).flatMap(lambda x: x).count()
✓ 0.4s
2052
```

Solution for Q2

(1)

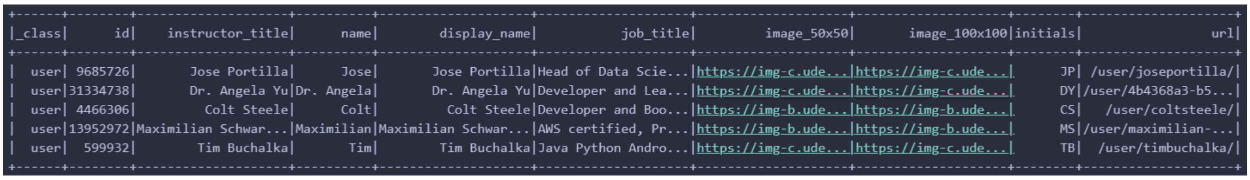
Firstly, read two `.csv` files by `spark.read.csv`, then rename some columns by `withColumnRenamed` as well as convert the data type of some columns if necessary according to the following questions.

```
course = spark.read.csv("data/Q2_data/courses.csv",header=True)
course = course.withColumnRenamed("title","course_title")
course = course.withColumn("created",to_timestamp("created", "yyyy-MM-dd'T'HH:mm:ss'Z'"))
```



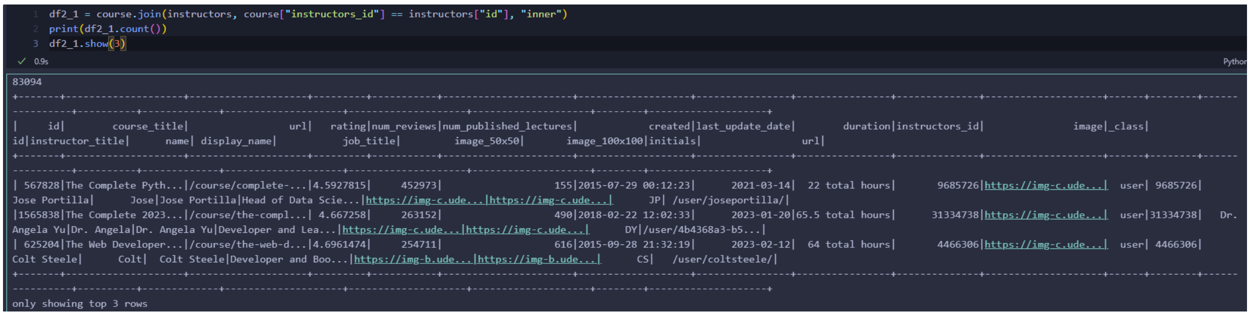
```
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+
|     id|        course_title|                 url|   rating|num_reviews|num_published_lectures|            created|last_update_date|        duration|instructors_id|               image|
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+
| 567828|The Complete Pyth...|/course/complete-...|4.5927815|     452973|                   155|2015-07-29 00:12:23|      2021-03-14|  22 total hours|       9685726|https://img-c.ude...|
|1565838|The Complete 2023...|/course/the-compl...| 4.667258|     263152|                   490|2018-02-22 12:02:33|      2023-01-20|65.5 total hours|      31334738|https://img-c.ude...|
| 625204|The Web Developer...|/course/the-web-d...|4.6961474|     254711|                   616|2015-09-28 21:32:19|      2023-02-12|  64 total hours|       4466306|https://img-c.ude...|
| 756150|Angular - The Com...|/course/the-compl...|4.5926924|     180257|                   472|2016-02-08 17:02:55|      2023-02-06|34.5 total hours|      13952972|https://img-c.ude...|
|2776760|100 Days of Code:...|/course/100-days-...|4.6952515|     177568|                   676|2020-01-24 10:47:21|      2022-11-30|  64 total hours|      31334738|https://img-c.ude...|
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+
only showing top 5 rows
```

```
instructors = spark.read.csv("data/Q2_data/instructors.csv",header=True)
instructors = instructors.withColumnRenamed("title","instructor_title")
```



```
+------+--------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------+--------------------+
|_class|      id|    instructor_title|      name|        display_name|           job_title|         image_50x50|       image_100x100|initials|                 url|
+------+--------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------+--------------------+
|  user| 9685726|       Jose Portilla|      Jose|       Jose Portilla|Head of Data Scie...|https://img-c.ude...|https://img-c.ude...|      JP| /user/joseportilla/|
|  user|31334738|       Dr. Angela Yu|Dr. Angela|       Dr. Angela Yu|Developer and Lea...|https://img-c.ude...|https://img-c.ude...|      DY|/user/4b4368a3-b5...|
|  user| 4466306|         Colt Steele|      Colt|         Colt Steele|Developer and Boo...|https://img-b.ude...|https://img-b.ude...|      CS|   /user/coltsteele/|
|  user|13952972|Maximilian Schwar...|Maximilian|Maximilian Schwar...|AWS certified, Pr...|https://img-b.ude...|https://img-b.ude...|      MS|/user/maximilian-...|
|  user|  599932|        Tim Buchalka|       Tim|        Tim Buchalka|Java Python Andro...|https://img-c.ude...|https://img-c.ude...|      TB|  /user/timbuchalka/|
+------+--------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------+--------------------+
```

Then `join` them together (inner join) by columns `instructor_id` and `course_id`.



```
df2_1 = course.join(instructors, course["instructors_id"] == instructors["id"], "inner")
print(df2_1.count())
df2_1.show(3)
✓ 0.9s
83094
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+------+------
--------+----------------+----------+-------------+--------------------+--------------------+--------------------+--------+--------------------+
|     id|        course_title|                 url|   rating|num_reviews|num_published_lectures|            created|last_update_date|        duration|instructors_id|               image|_class|
id|instructor_title|      name| display_name|           job_title|         image_50x50|       image_100x100|initials|                 url|
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+------+------
--------+----------------+----------+-------------+--------------------+--------------------+--------------------+--------+--------------------+
| 567828|The Complete Pyth...|/course/complete-...|4.5927815|     452973|                   155|2015-07-29 00:12:23|      2021-03-14|  22 total hours|       9685726|https://img-c.ude...|  user| 9685726|
Jose Portilla|      Jose|Jose Portilla|Head of Data Scie...|https://img-c.ude...|https://img-c.ude...|      JP| /user/joseportilla/|
|1565838|The Complete 2023...|/course/the-compl...| 4.667258|     263152|                   490|2018-02-22 12:02:33|      2023-01-20|65.5 total hours|      31334738|https://img-c.ude...|  user|31334738|   Dr.
Angela Yu|Dr. Angela|Dr. Angela Yu|Developer and Lea...|https://img-c.ude...|https://img-c.ude...|      DY|/user/4b4368a3-b5...|
| 625204|The Web Developer...|/course/the-web-d...|4.6961474|     254711|                   616|2015-09-28 21:32:19|      2023-02-12|  64 total hours|       4466306|https://img-c.ude...|  user| 4466306|
Colt Steele|      Colt|  Colt Steele|Developer and Boo...|https://img-b.ude...|https://img-b.ude...|      CS|   /user/coltsteele/|
+-------+--------------------+--------------------+---------+-----------+----------------------+-------------------+----------------+----------------+--------------+--------------------+------+------
--------+----------------+----------+-------------+--------------------+--------------------+--------------------+--------+--------------------+
only showing top 3 rows
```

(2)

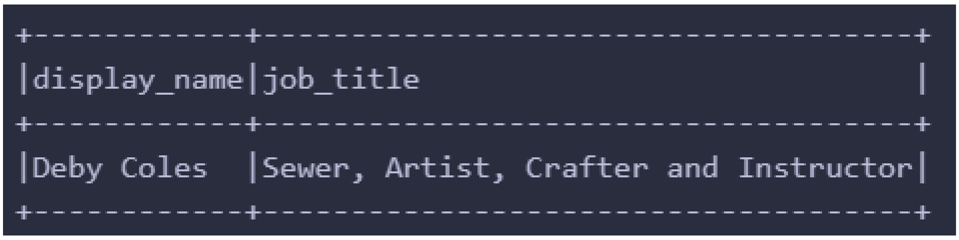
Before selecting by `spark SQL`, a temporary view should be created by `createOrReplaceTempView`.

Here is the detail about the SQL query referring to the requirements:

- **Highest course rating**: order the data by `rating` in descending order and select the first row.
- **Among all courses that are related to 'spark'**: use `like` in `where` to filter the rows that contain 'spark' in the column `course_title`.

- **Created after 2018-01-01 00:00:00**: select the rows that the column `created` is later than `2018-01-01 00:00:00` in `where`.

```
df2_1.createOrReplaceTempView("df2_1")
spark.sql("select display_name, job_title from df2_1 where course_title like '%spark%' and created > '2018-01-01 00:00:00' order by rating desc LIMIT 1").show(truncate=False)
```



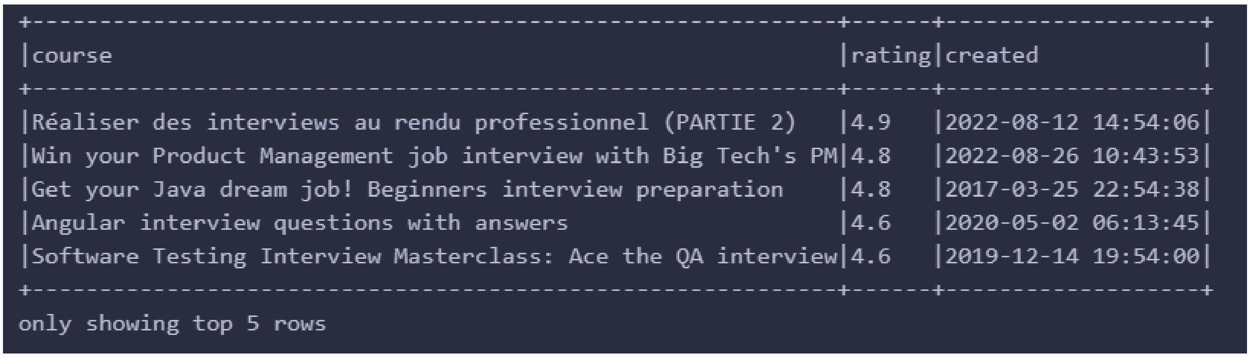
```
+------------+------------------------------------+
|display_name|job_title                           |
+------------+------------------------------------+
|Deby Coles  |Sewer, Artist, Crafter and Instructor|
+------------+------------------------------------+
```

(3)

Here is a brief explanation of the SQL query concerning the problem description:

- **All courses that are (a) related to 'interview'**: As `like` is case-insensitive, we can use `%interview%` to match any string that contains 'interview', which can also include 'interviews'.
- **Sorted by course_rating in descending order and created in descending order (newest first)**: order the data by `rating` in descending order and `created` in descending order.
- **Course rating should be firstly rounded to one decimal place**: use `round` to round the column `rating` to one decimal place.

```
course.createOrReplaceTempView("course")
spark.sql("select course_title  as course,round(rating,1) as rating , created from course where course_title like '%interview%' order by rating desc, created desc").show(5,truncate=False)
```



```
+------------------------------------------------------------+------+-------------------+
|course                                                      |rating|created            |
+------------------------------------------------------------+------+-------------------+
|Réaliser des interviews au rendu professionnel (PARTIE 2)   |4.9   |2022-08-12 14:54:06|
|Win your Product Management job interview with Big Tech's PM|4.8   |2022-08-26 10:43:53|
|Get your Java dream job! Beginners interview preparation    |4.8   |2017-03-25 22:54:38|
|Angular interview questions with answers                    |4.6   |2020-05-02 06:13:45|
|Software Testing Interview Masterclass: Ace the QA interview|4.6   |2019-12-14 19:54:00|
+------------------------------------------------------------+------+-------------------+
only showing top 5 rows
```