ANALYSIS OF CLINICAL TRIAL DATA (MAY 2022)

(1.) **DESCRIPTION OF SET UP:**

This project involves the analysis of given files containing clinical trial data sets, a list of pharmaceutical industries, and a condition hierarchy of information.

The purpose is to analyze the given data and gain further insight into clinical trial data sets for the year 2021. A clinical trial is a kind of research designed to evaluate and test new interventions such as psychotherapy or medications. The usage of big data tools and techniques will be employed to perform various analysis of the three main data files given.

(1a.)**Description of the files to be used include:**

(1a.)**Clinicaltrial_2021.csv.gz**

This file was compressed into .gz file from the source. It means an archive compressed by the standard GNU zip(gzip) compression algorithm. It needs to be decompressed or unzipped to view the contents. This will be decompressed in the data cleaning and preparation process.
(Source: ClinicalTrials.gov)

(1a.) **Mesh.csv**

This is a plain text file that contains a list of data and allows the data to be stored in a table of structured data. It uses the comma character to separate (or delimit data). This file can be opened in any program hence it is suitable for use in Databricks.
(Source: U.S. National Library of Medicine.)

(1a.)**Pharma.csv**

This file shares the same features as the mesh.csv file above.

(Source: https://violationtracker. goodjobsfirst.org/industry/pharmaceuticals)

(1b). **Description of the tool used:**

The tool used is called Databricks, a cloud-based data engineering tool that is widely used by companies to process and transform large quantities of data and explore the data. Databricks is developed by the creators of pyspark, an interface for Apache spark in python. It makes available Pyspark shells for a thorough analysis of data in a computing environment where various components are spread across multiple computers.

(1c.) **Insight of Analysis:**

The analysis bothers on the number of studies in the dataset, all the types of studies along with their frequencies, the top five conditions, and their frequencies, the five most frequent roots, the ten most commons sponsors that are not pharmaceuticals together with the clinical trials sponsored by them, and the number of completed studies each month in the year 2021.

(1d.) **Further Analysis :**

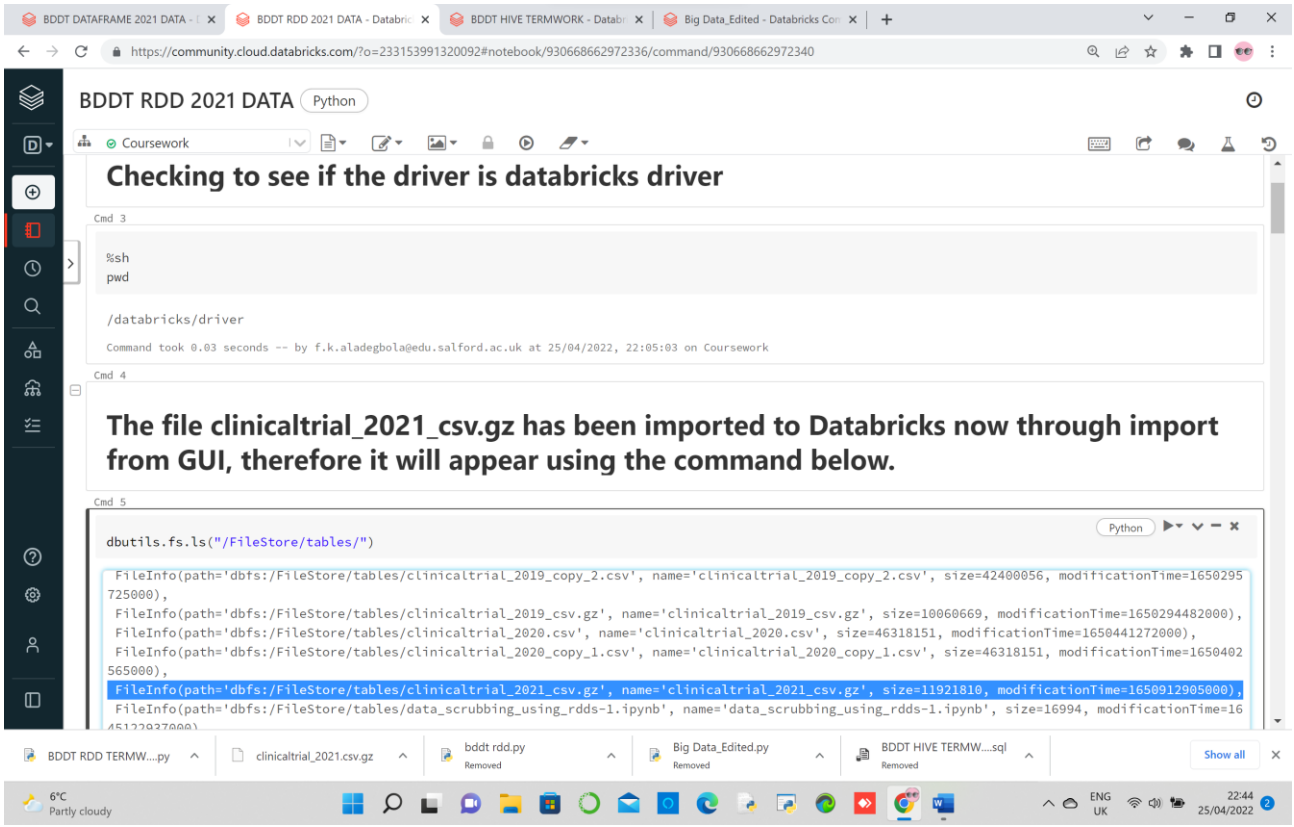
A comparison between the number of completed and uncompleted studies, the top ten interventions with their frequencies, and the first five Sponsors in the year 2021 with the highest number of sponsored clinical trials were analyzed.

(2.) **DATA CLEANING AND PREPARATION**

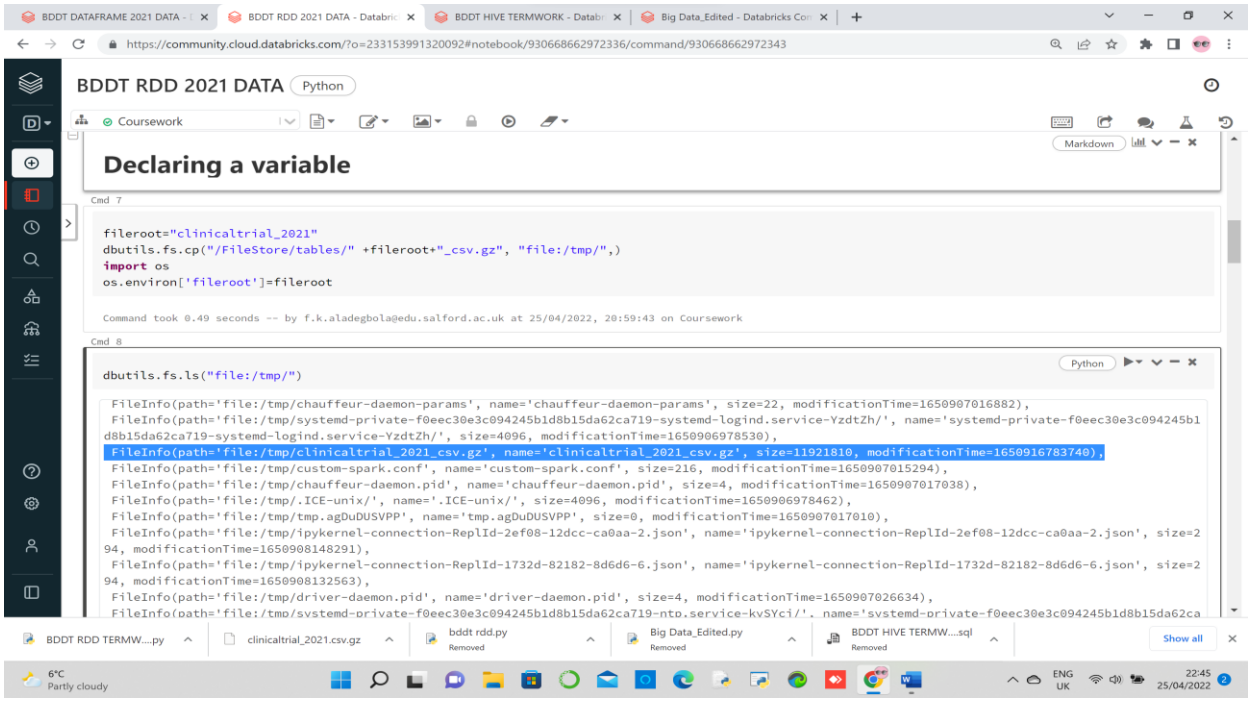
This is the process of ensuring that data is correct, consistent, and usable.

(2.1)**Description of Unzipping of Clinicaltrial_2021.csv.gz**

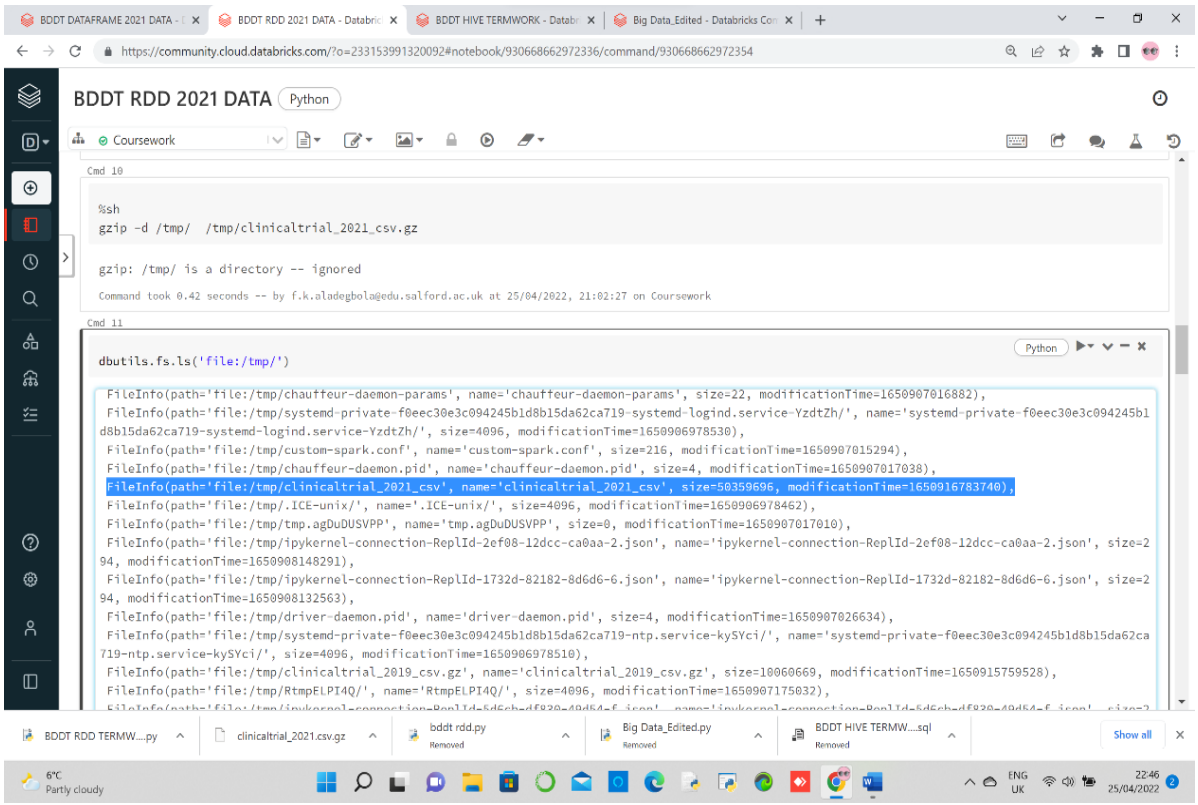
The databricks environment was checked to see if the driver is a databricks driver, then the imported file clinicaltrial_2021_csv.gz was imported to Databricks through import from the GUI(Graphic User Interface). This was confirmed present in FileStore/tables with dbutils.fs.ls command as below:



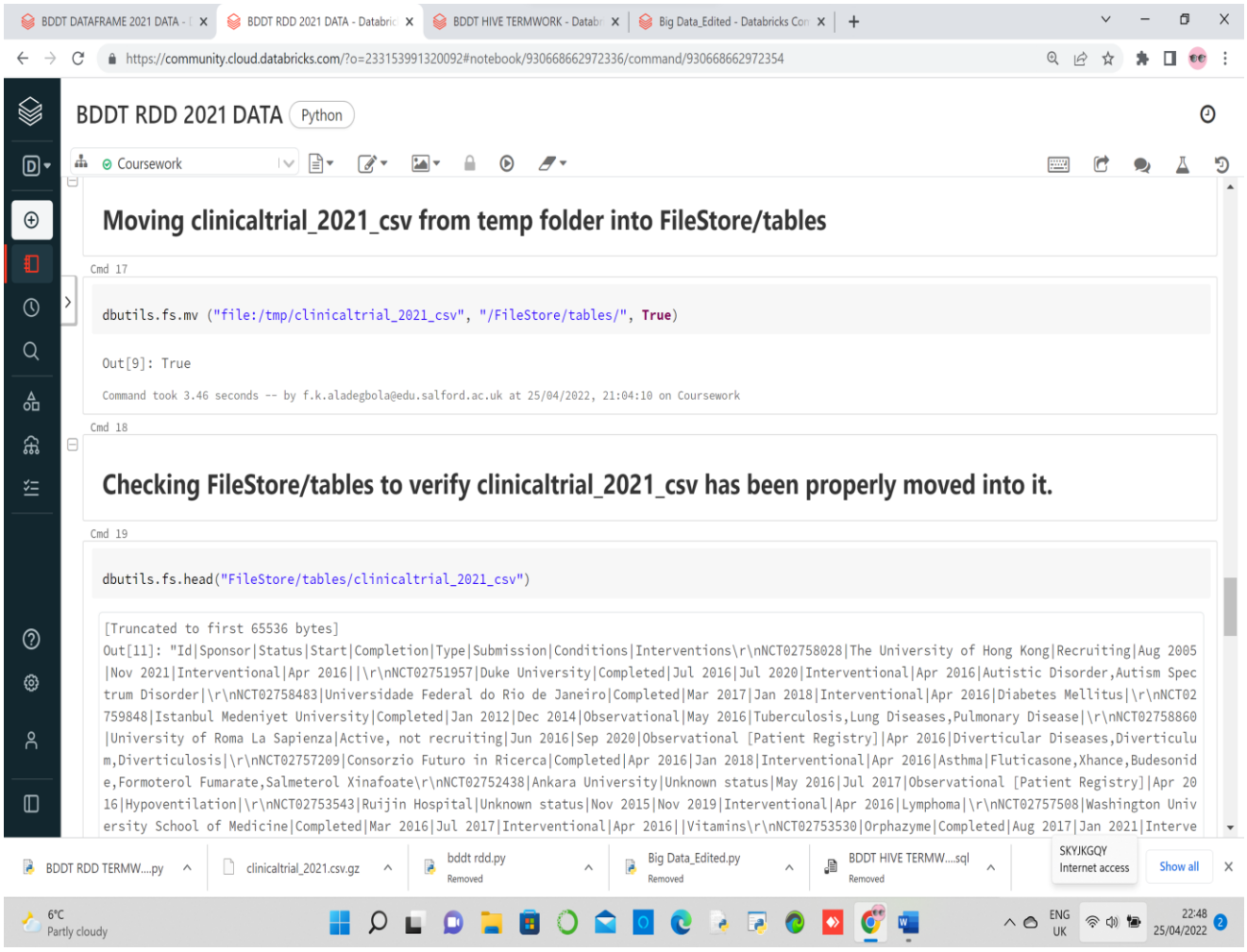
(2.2) A variable called fileroot was declared as clinical_2021



(2.3) The file Clinicaltrial_2021_csv.gz was unzipped with the command in the screenshot below:

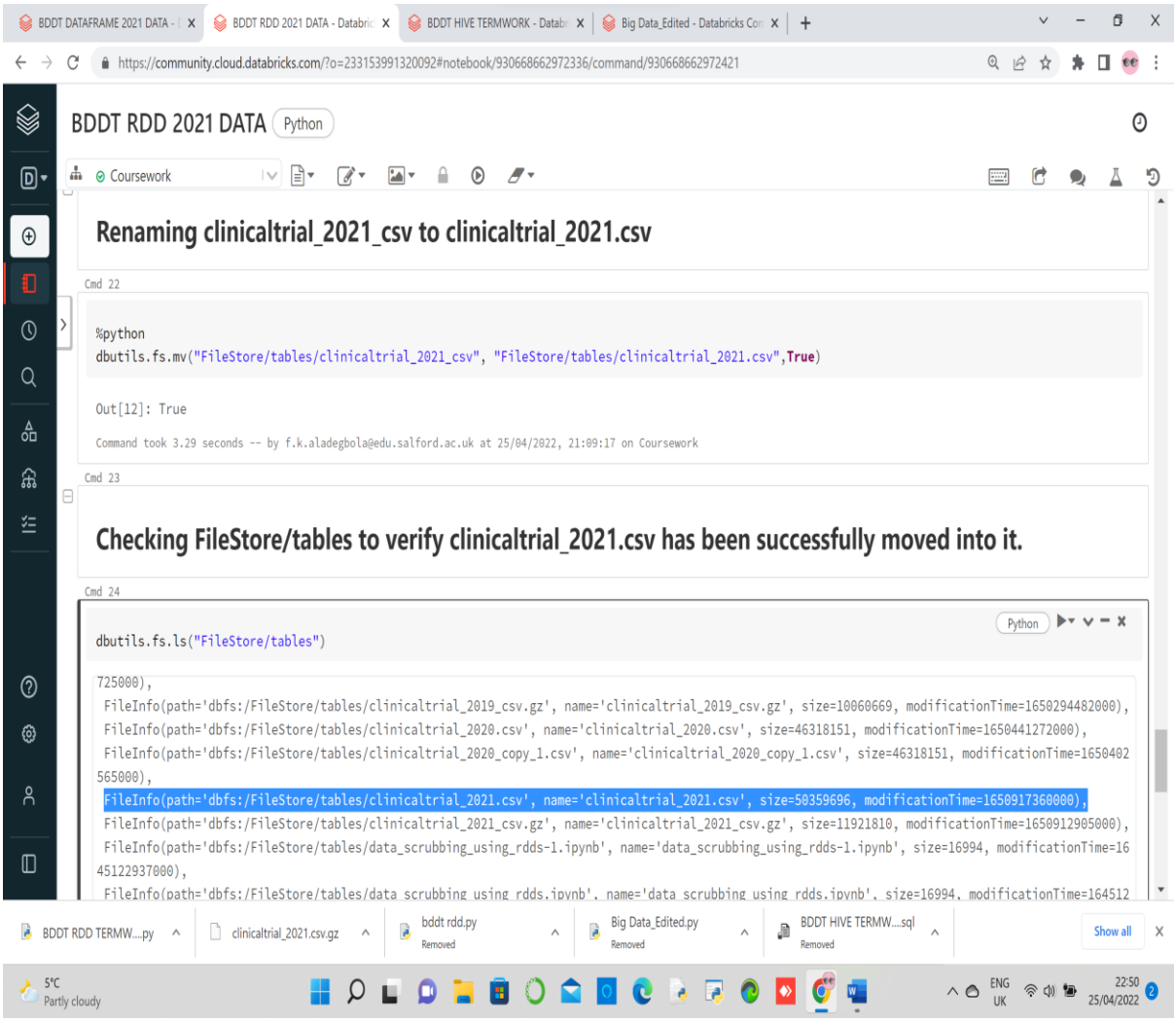


(2.4) The unzipped file which has now become clinical_2021_csv was moved from temp folder to FileStore/tables using the dbutils.fs.mv command.



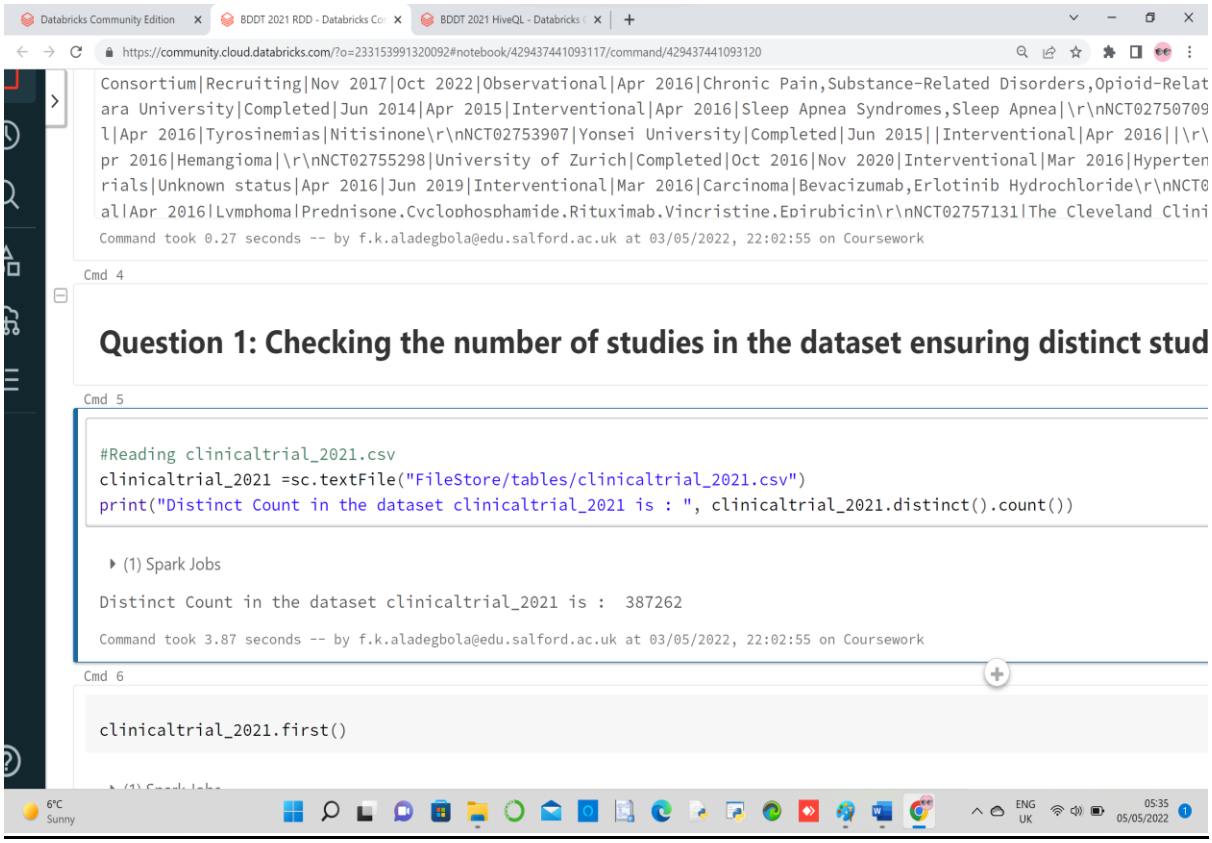
(2.5) The file clinicaltrial_2021_csv was renamed to clinicaltrial_2021.csv by using the dbutils.fs.mv command. This moves clinicaltrial_2021_csv from its present location to FileStore/tables with the new name clinicaltrail_2021.csv

(2.6) The accuracy was checked with the command dbutils.fs.ls(“FileStore/tables”) as shown in the screenshot below

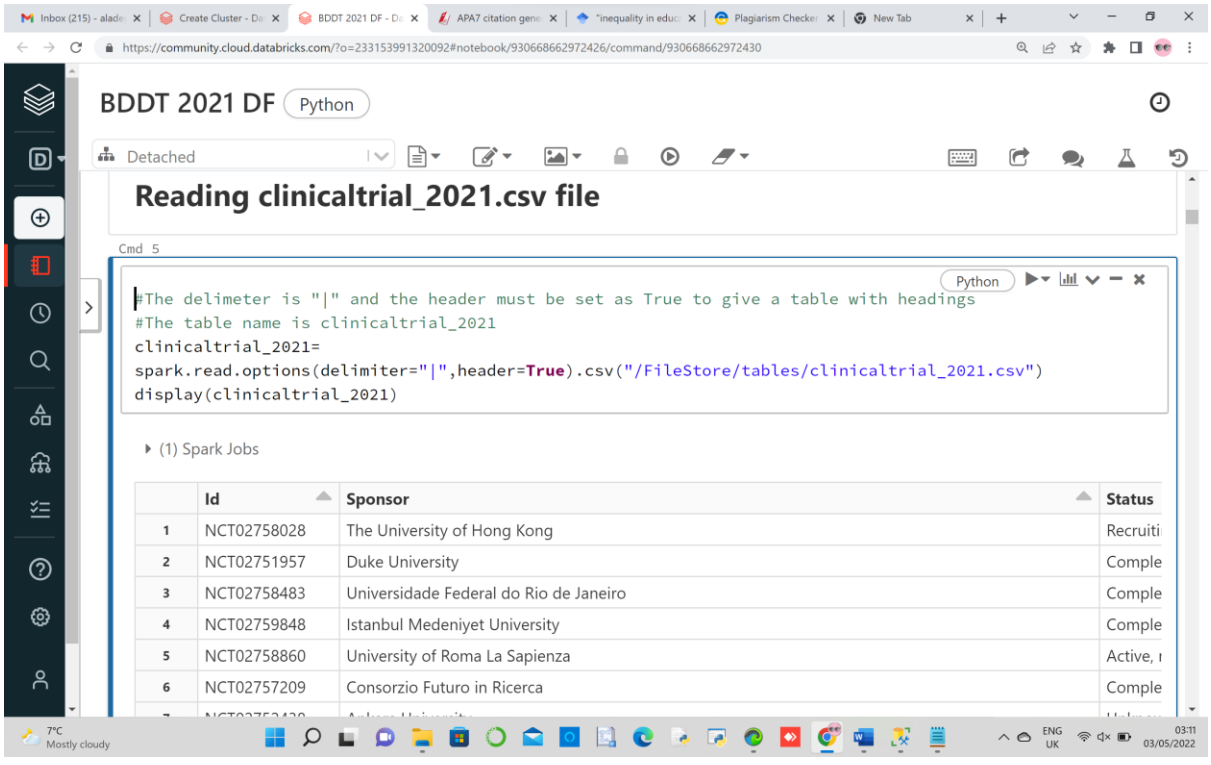


(2.7) Opening the files in Databricks:

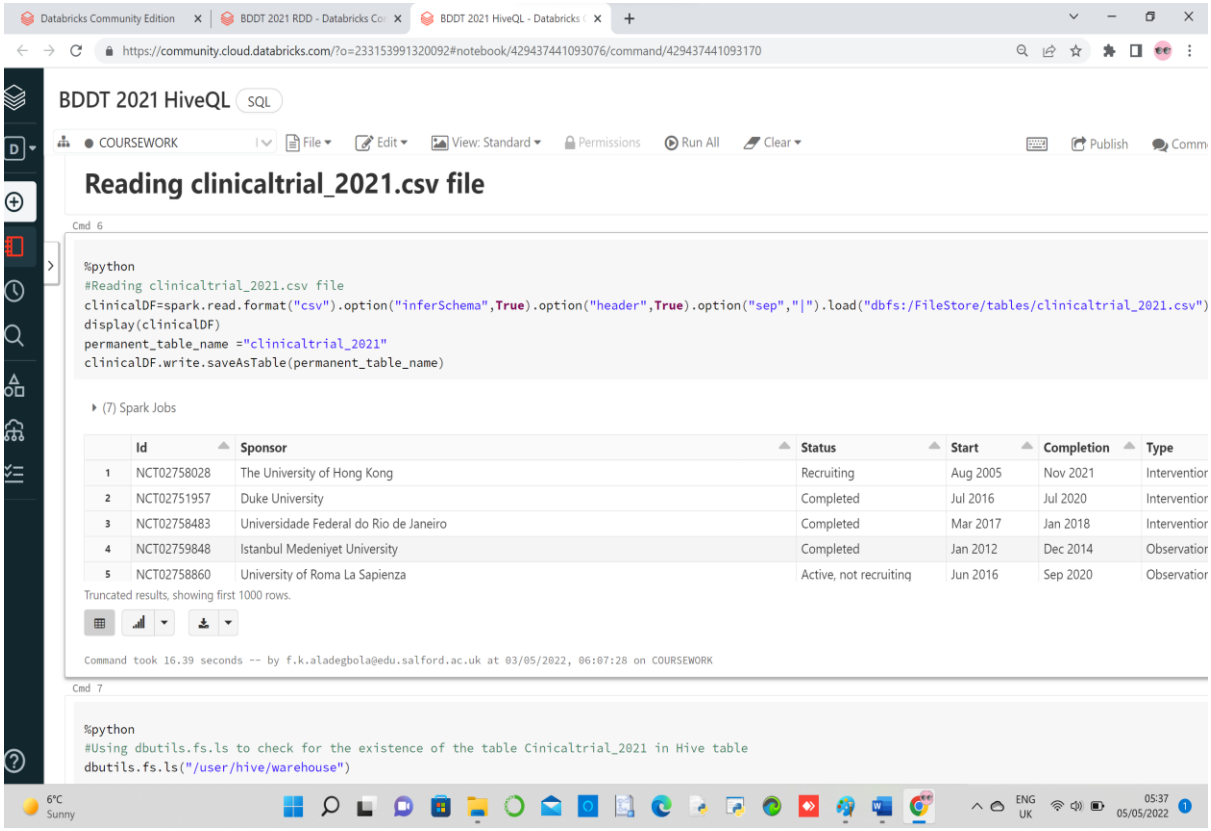
clinicaltrial_2021.csv file is read by using: sc.textfile in rdd



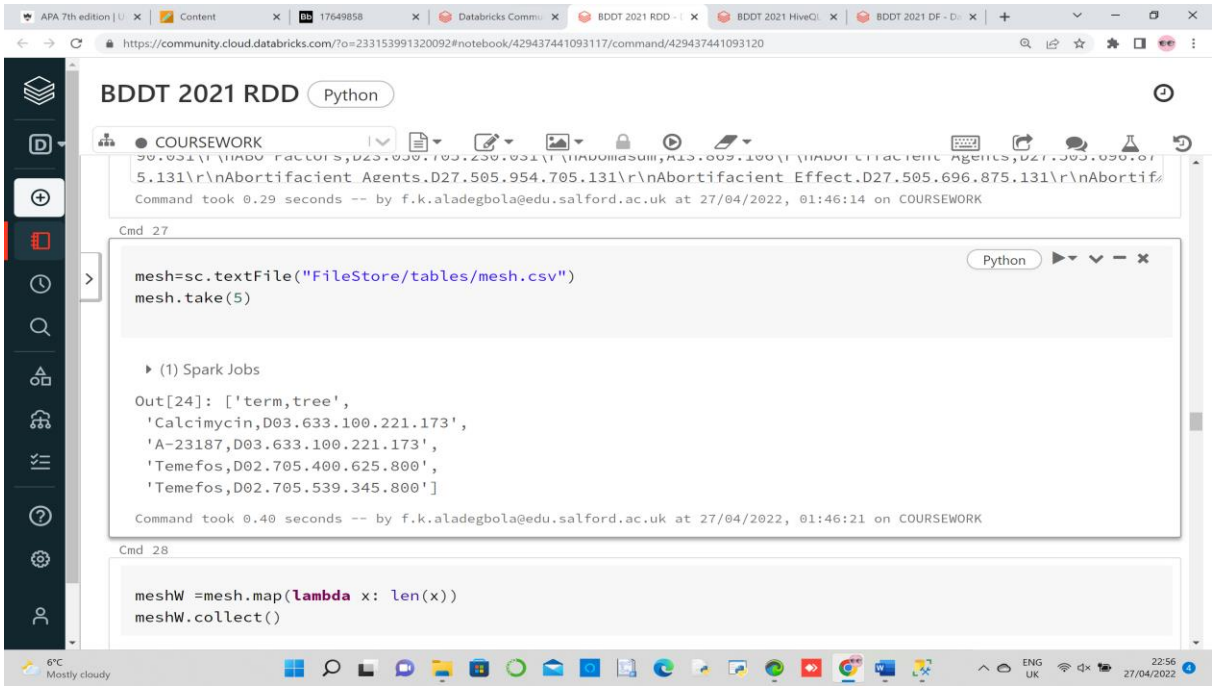
Spark.read.options in DataFrames



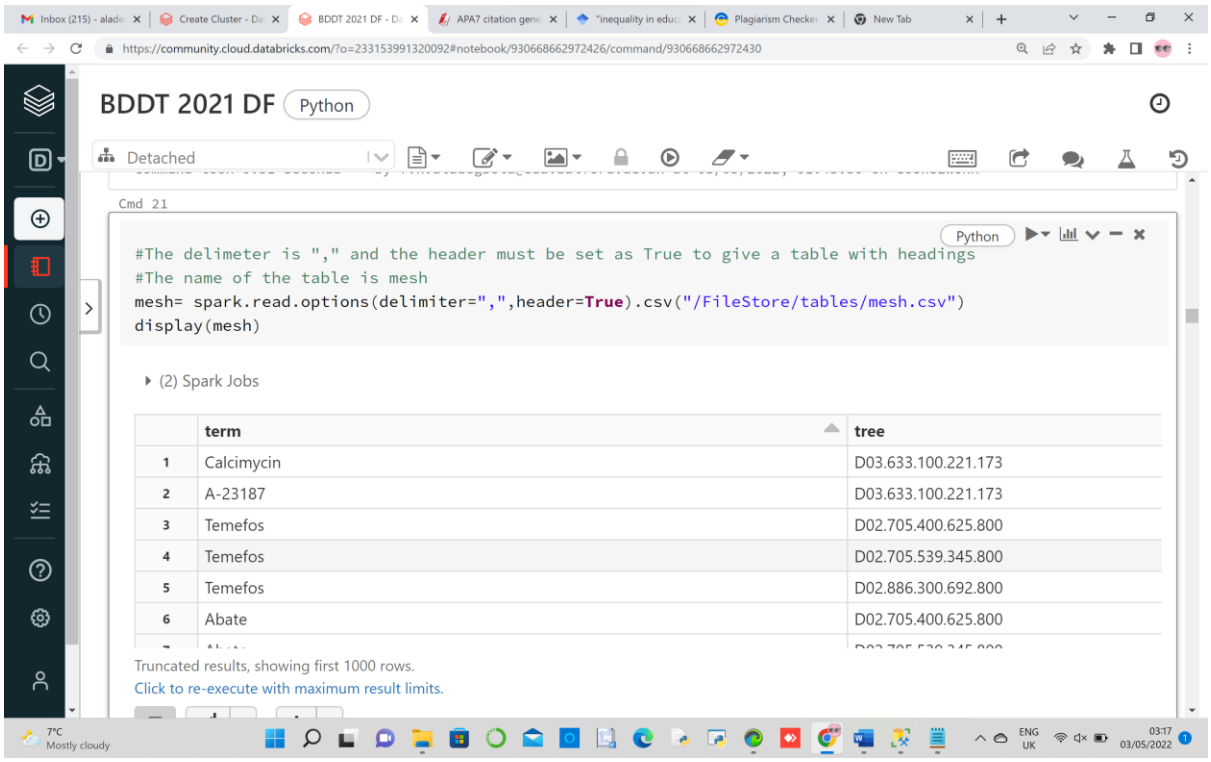
Spark.read.options declaring infer schema as True and setting delimiters in Hivegl



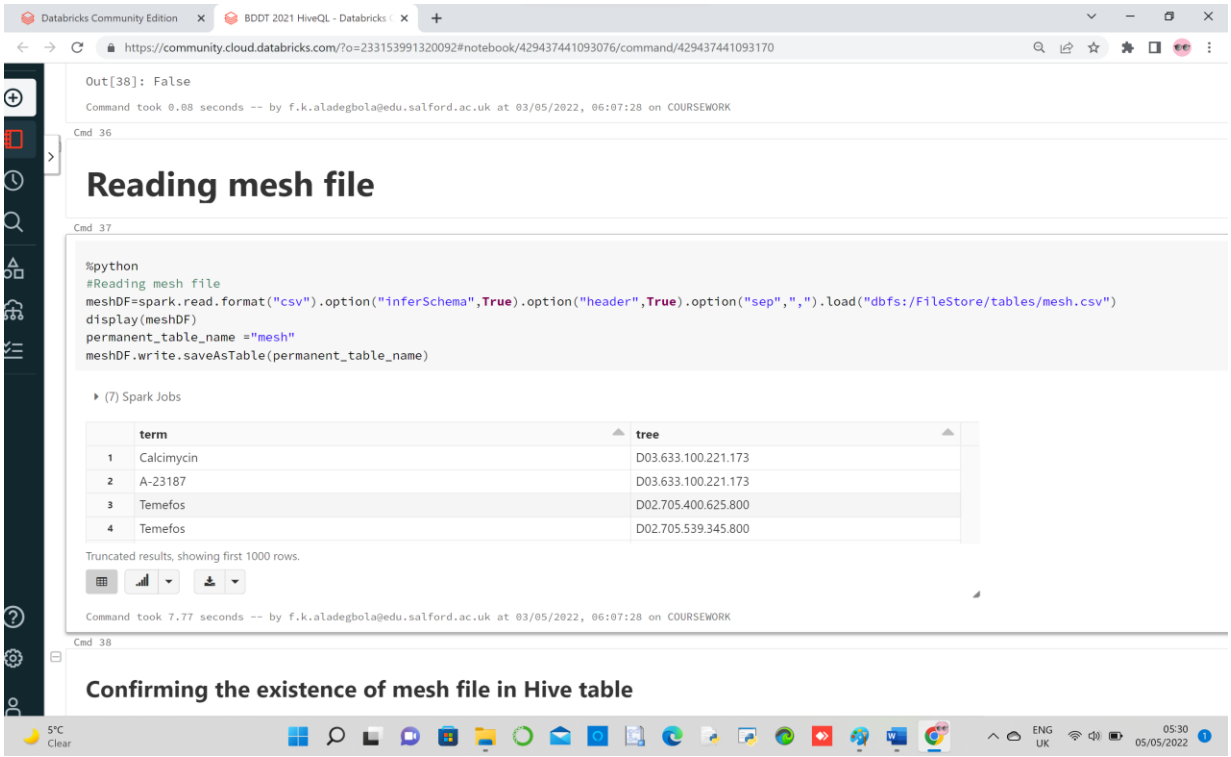
Reading Mesh.csv file : sc.textfile in rdd



Spark.read.options in DataFrames

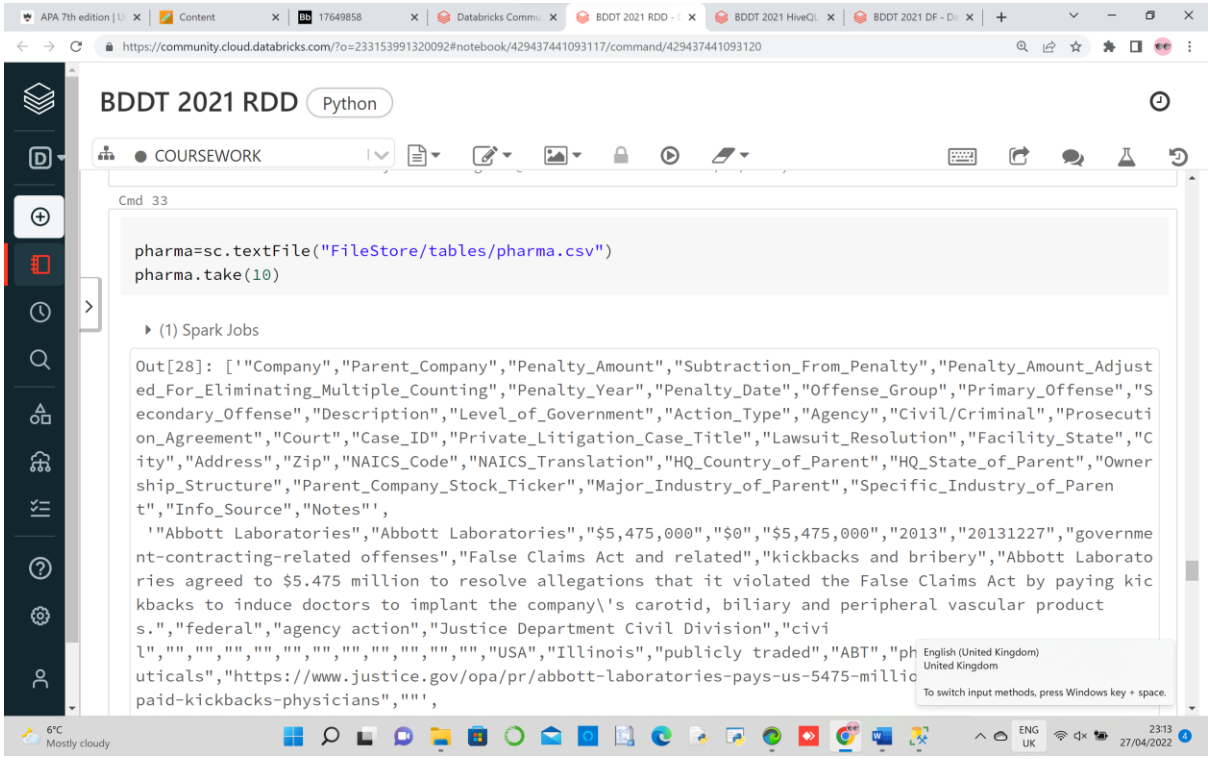


Spark.read.options declaring infer schema as True and setting delimiters in Hivegl

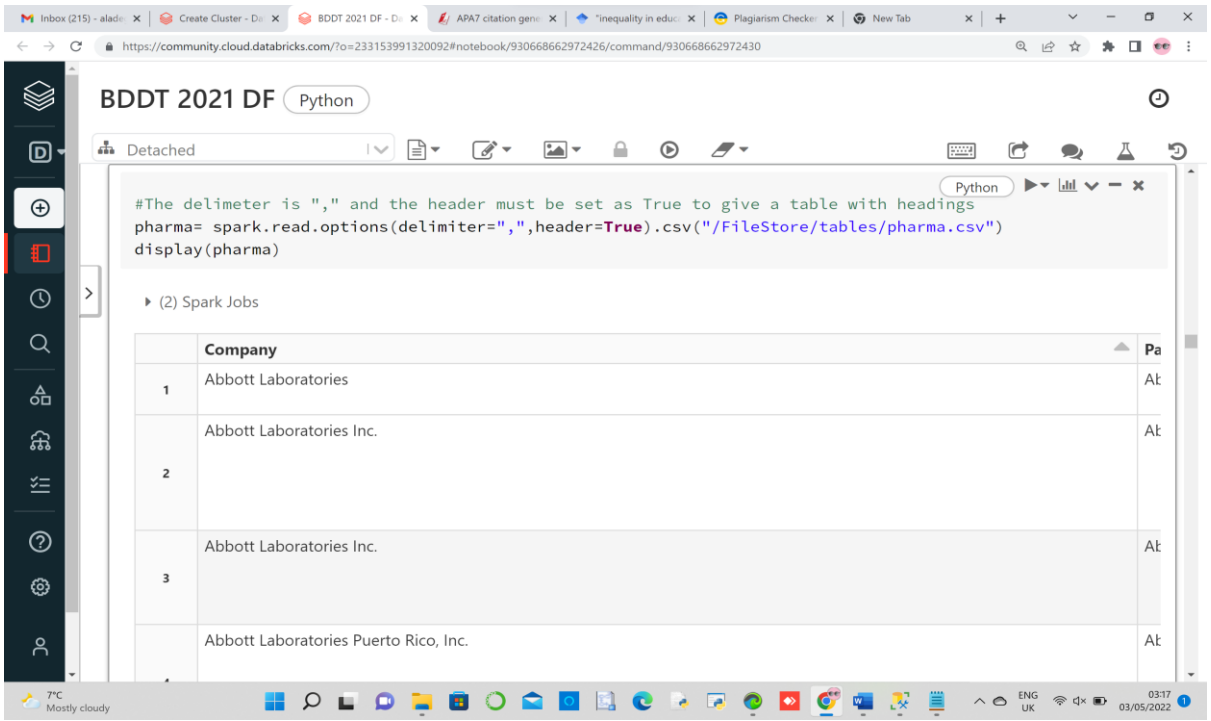


Pharma.csv

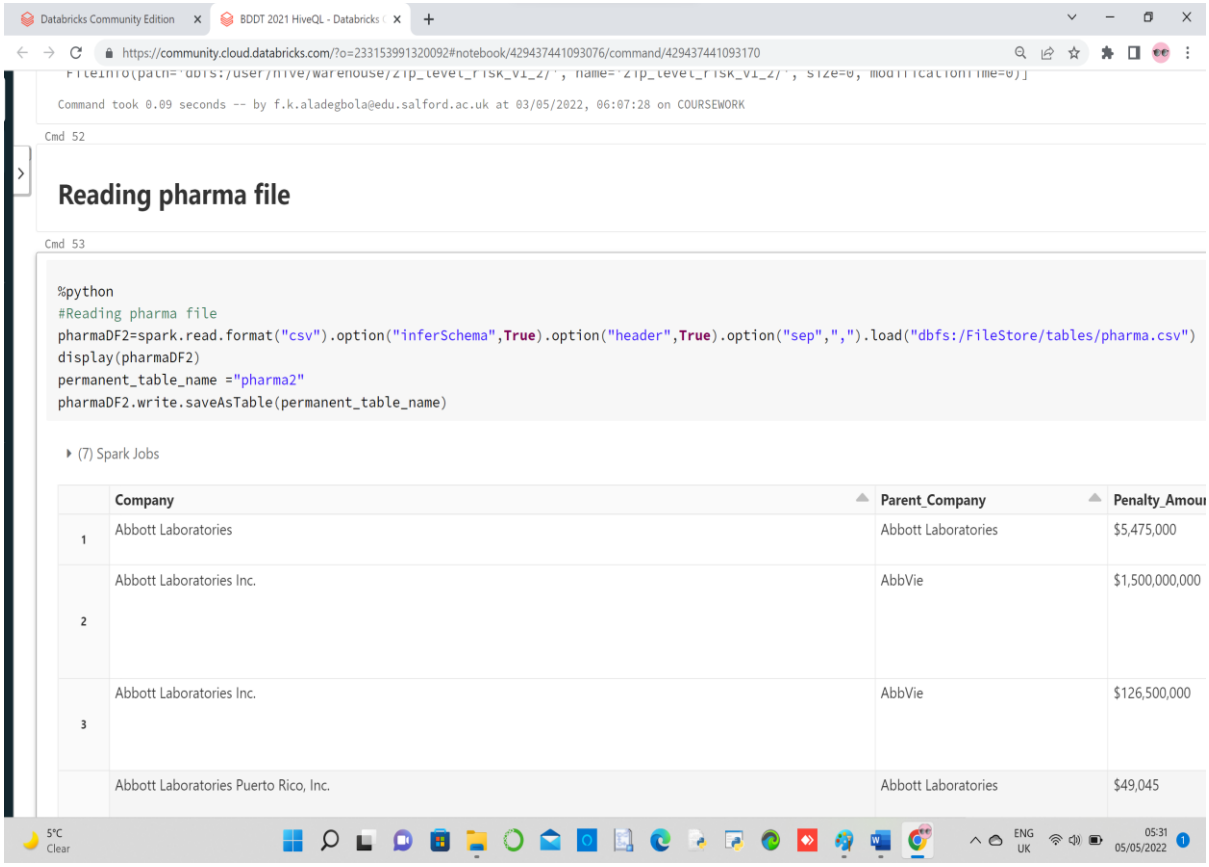
sc.textfile in rdd



Spark.read.options in DataFrames



Spark.read.options declaring infer schema as True and setting delimiters in Hiveql



3. PROBLEM ANSWERS

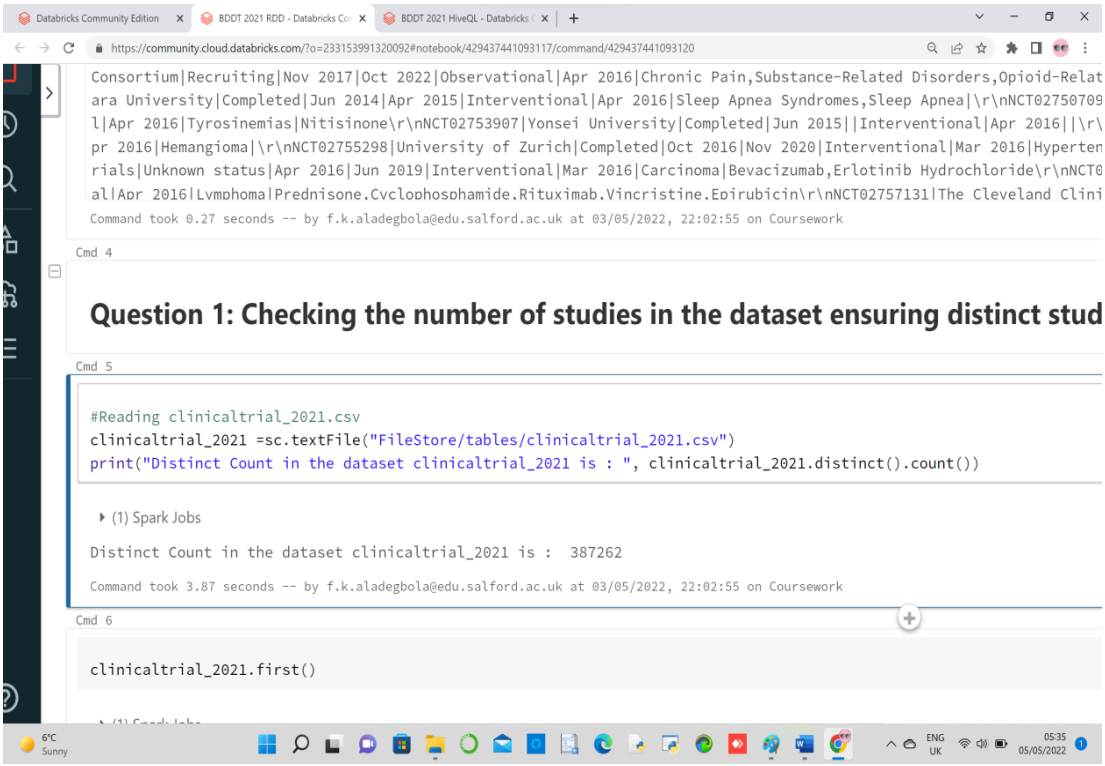
(a) Question 1 : The number of studies in the dataset?

- **Assumptions made:**

  It was assumed that the dataset contained all the clinical trials conducted for the year 2021 and there was no error in recording.

(3a.) **Implementation outline RDD**

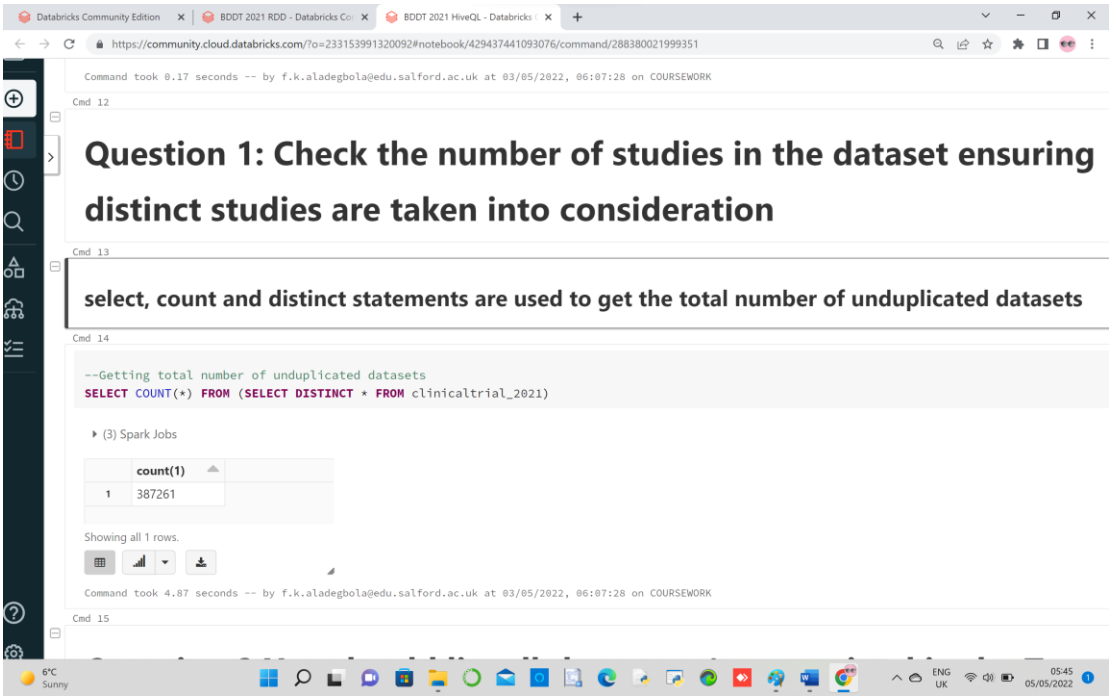
To erase the possibility of counting a clinical study twice, the distinct function was used in counting to remove duplicated values.



Removal of the data heading called header is essential because it was counted with the clinical trial. This was achieved with the filter function by passing it through the first rdd . The resultant rdd called headerRemvRdd1 was thereafter used in the final count.

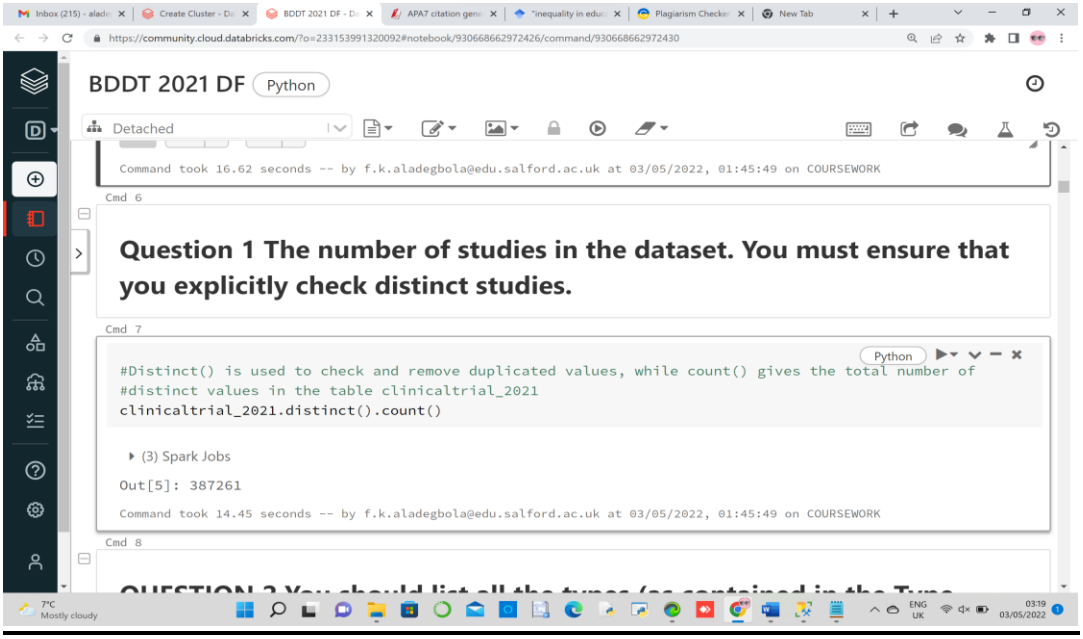
(3a.) **HiveQL implementation outline**

Opening Clinicaltrial_2019 .csv file transformed it into a table of structured data. To get the count in hiveql, select distinct and count statement is used to return select unduplicated rows with their count respectively.



(3a.) **PySpark implementation outline DF**

Dataframe also makes use of distinct and count statements to return non-duplicated rows. Unlike hiveql, it is applied directly to the named dataframe used to open the file.



(3a.) **Result on the submission (final) data –**

The number of studies in clinicaltrial_2019.csv resulted in 387,261using the three implementations in databricks.

(3a.) **Discussion of result:**

The number of studies in 2019 and 2020 was 326348 and 356466 respectively. Comparing this number to the number in 2021, a total of 387,261 shows an increase in the number of clinical trials being carried out yearly.

(3b.) **PROBLEM ANSWERS**

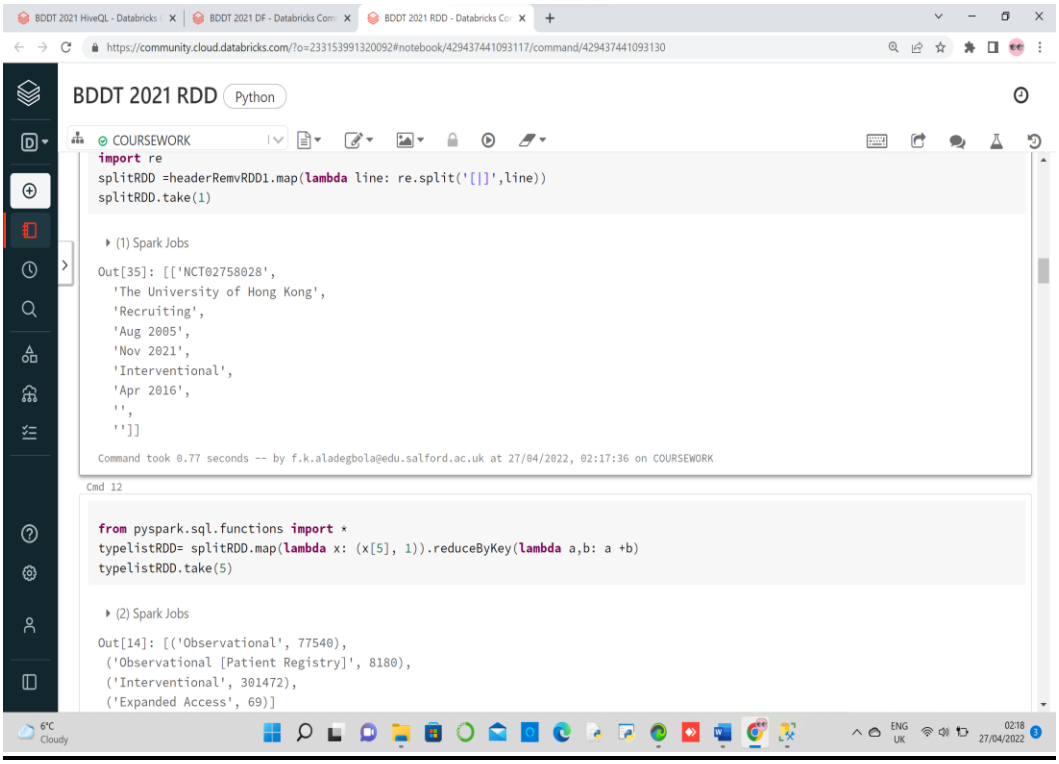
- **Question 2:** You should list all the types (as contained in the Type column) of studies in the dataset along with the frequencies of each type. These should be ordered from most frequent to least frequent.

  (3b.) **Assumptions made:**

  It was assumed that four types of studies were carried out in the year 2021 and all the studies carried out were properly documented.

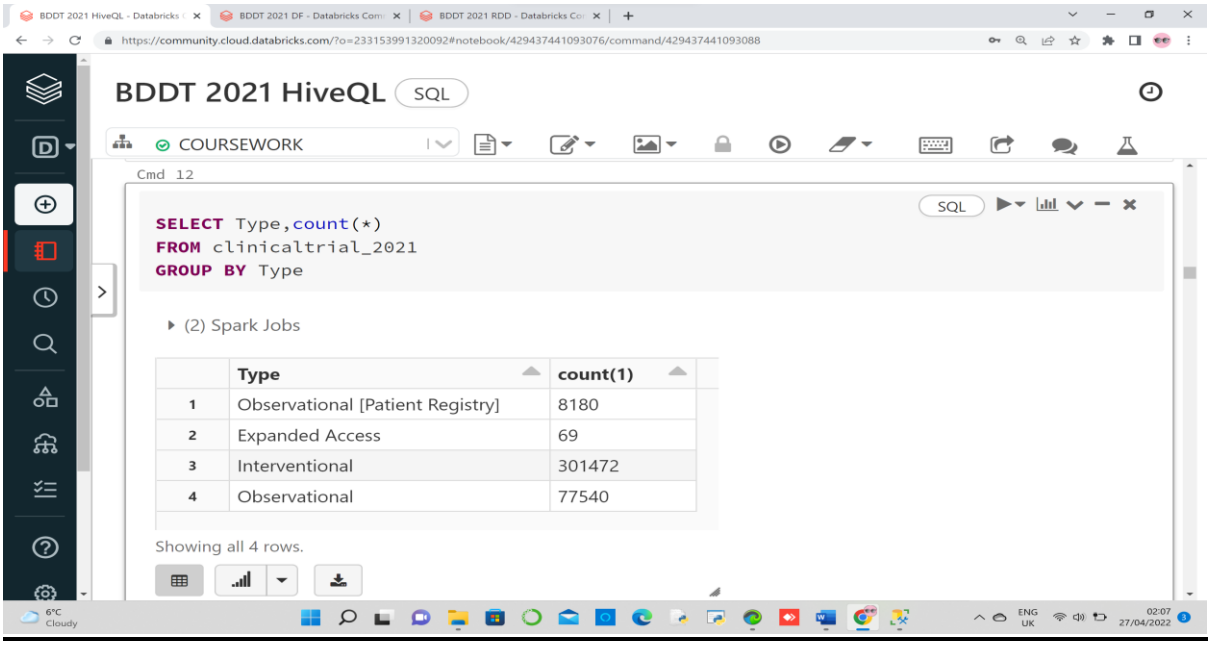
  (3b.) **PySpark implementation outline RDD**

  The rdd used in question one called headerRemvRDd1, whose header was removed was used to map a new rdd called splitRDD. Re was imported as a library in order for the function split to work with the anonymous function lambda. Splitting the file made each row a list. The first column in each row represents x[0], and the second x[1] e.t.c The Type of study is the 6^{th} column represented by x[5]. This was mapped into a new rdd called typelistRdd with the reduceByKey which assigns the number one to each type of clinical trial and then aggregates it and returns the total count for each type of clinical trial.



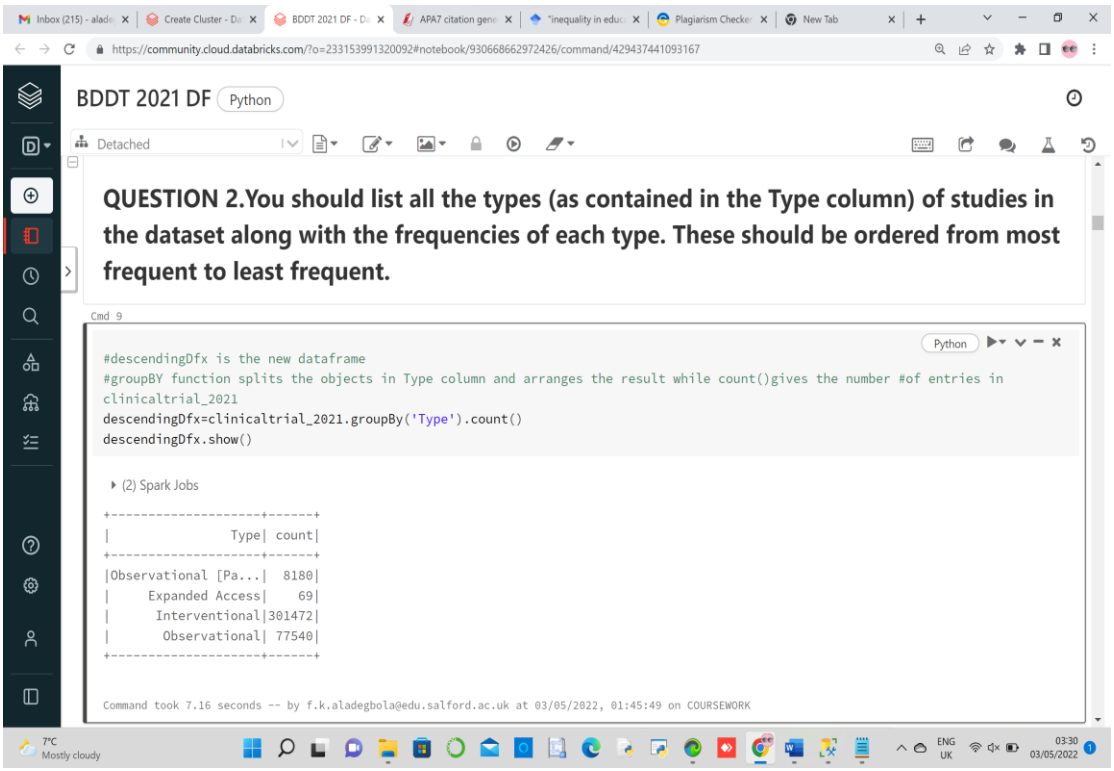
(3b.) **HiveQL implementation outline**

Clinicaltrial_2021 is a structured table with defined rows and columns. In order to get the total number of the Types of clinical trials, the ‘select’ statement with ‘count’ was used to specifically return the Type column and enumerate each of the Types. The ‘‘groupby’ statement was then used to aggregate the total number of counts for each type of clinical trial.



(3b.) **PySpark implementation outline DF**

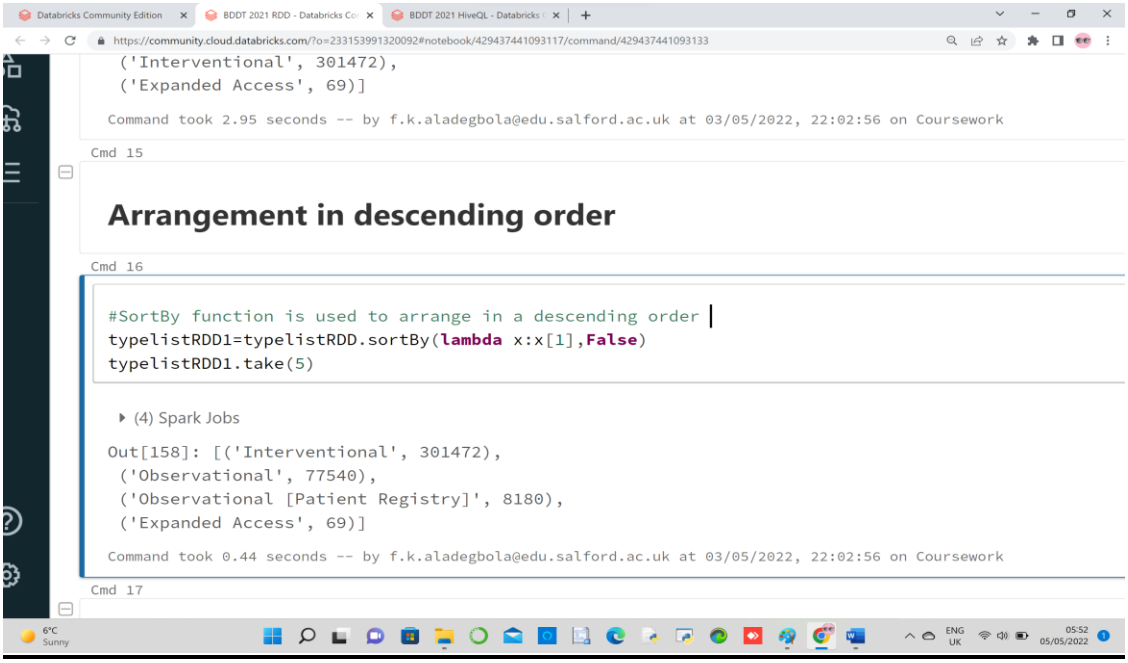
The Types of the clinical trial was obtained by applying ‘groupBy()’ and ‘count()’ statements to type column. This groups rows that have the same values into summary rows and the result is set by the count column.



(3b.) **Result on the submission (final) data**

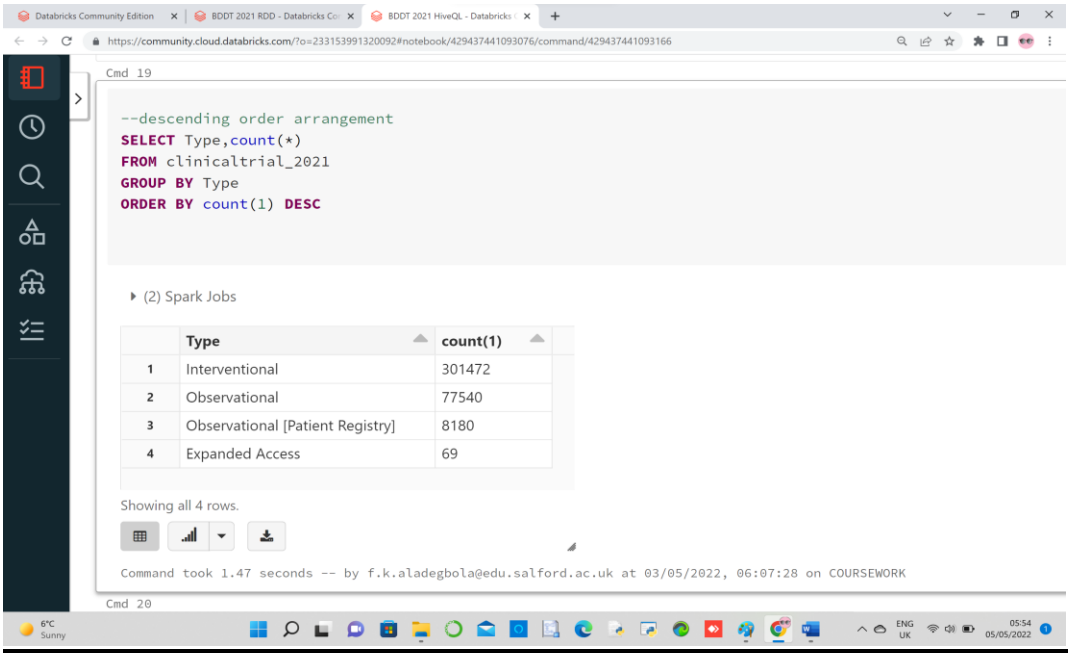
RDD RESULT

The result obtained was arranged in descending order by making use of the anonymous function lambda and sortBy including the Boolean value False to show the descending order in another rdd called typelistRDD1.



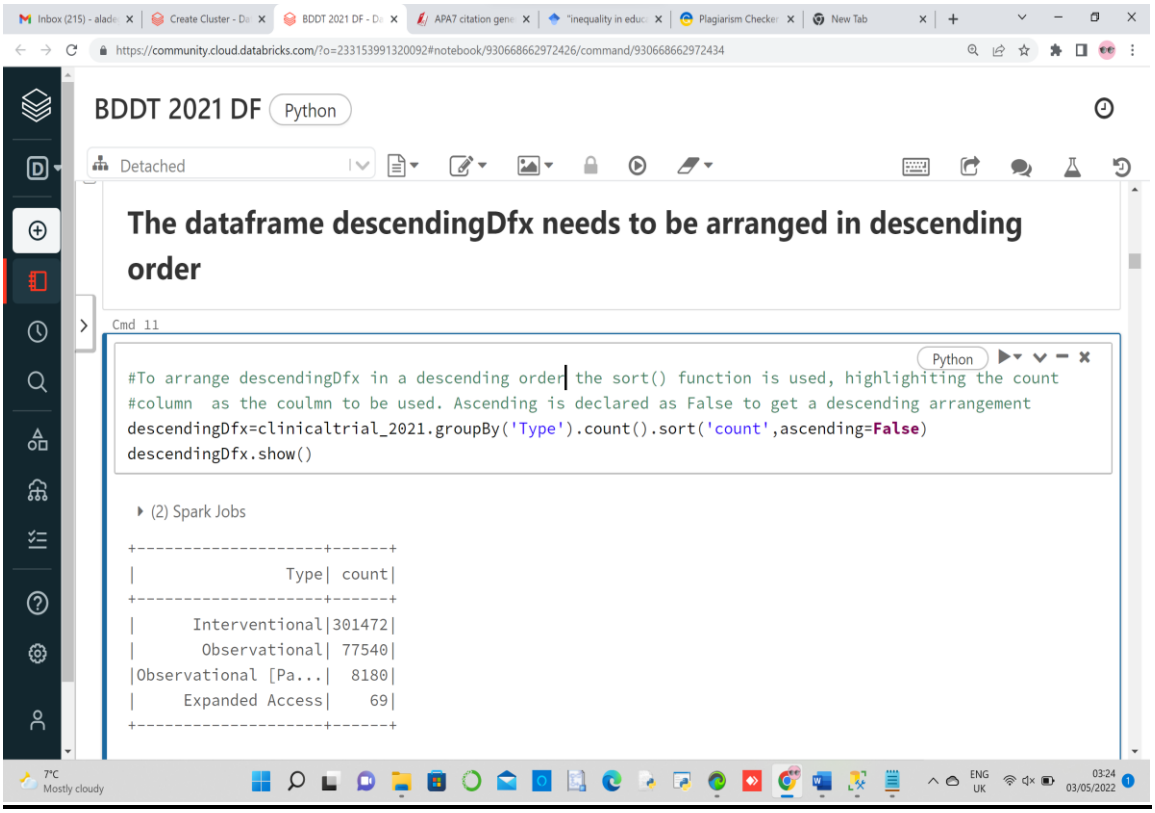
(3b.) **HIVEQL RESULT**

This implementation used the statements ‘orderby’ and ‘Desc’ meaning descending to list the types of clinical trials and arrange their frequencies in descending order.



(3b.) **DATAFRAME RESULT**

ascending=False in sort was adopted for descending order arrangement.



(3b.) **Discussion of result:**

Using the three implementations on data bricks, interventional emerged as the most widely used type of clinical trial and expanded access the least. Also, this result shows that the types of clinical trials used have remained the same for the years 2019,2020, and 2021 with no addition of other Types.

(3c.) **PROBLEM ANSWERS**

Question 3: The top 5 conditions (from Conditions) with their frequencies

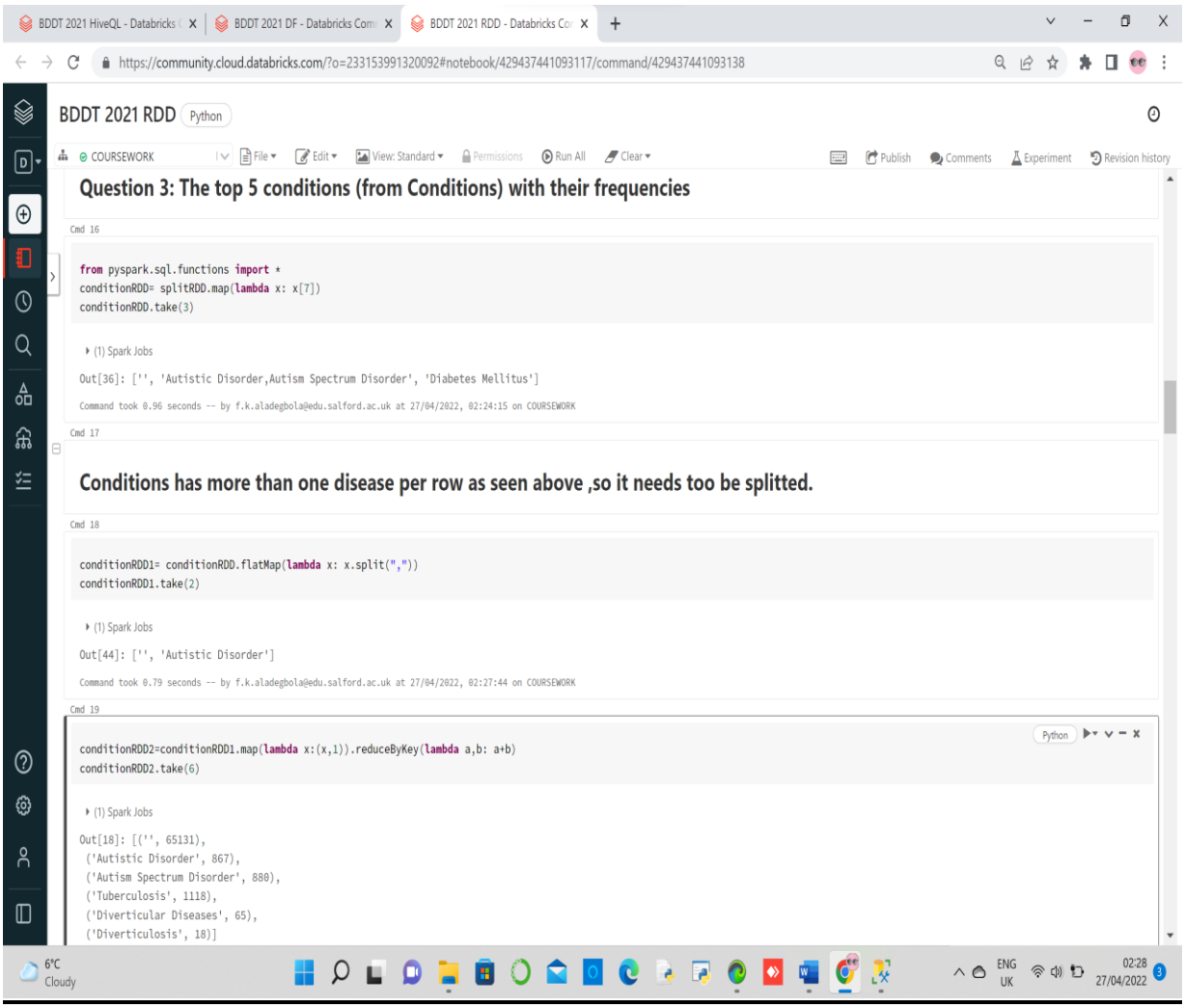
- **Assumptions made:**

  The assumption was that the data was properly documented and free of errors during collation.

(3c.) **PySpark implementation outline RDD**

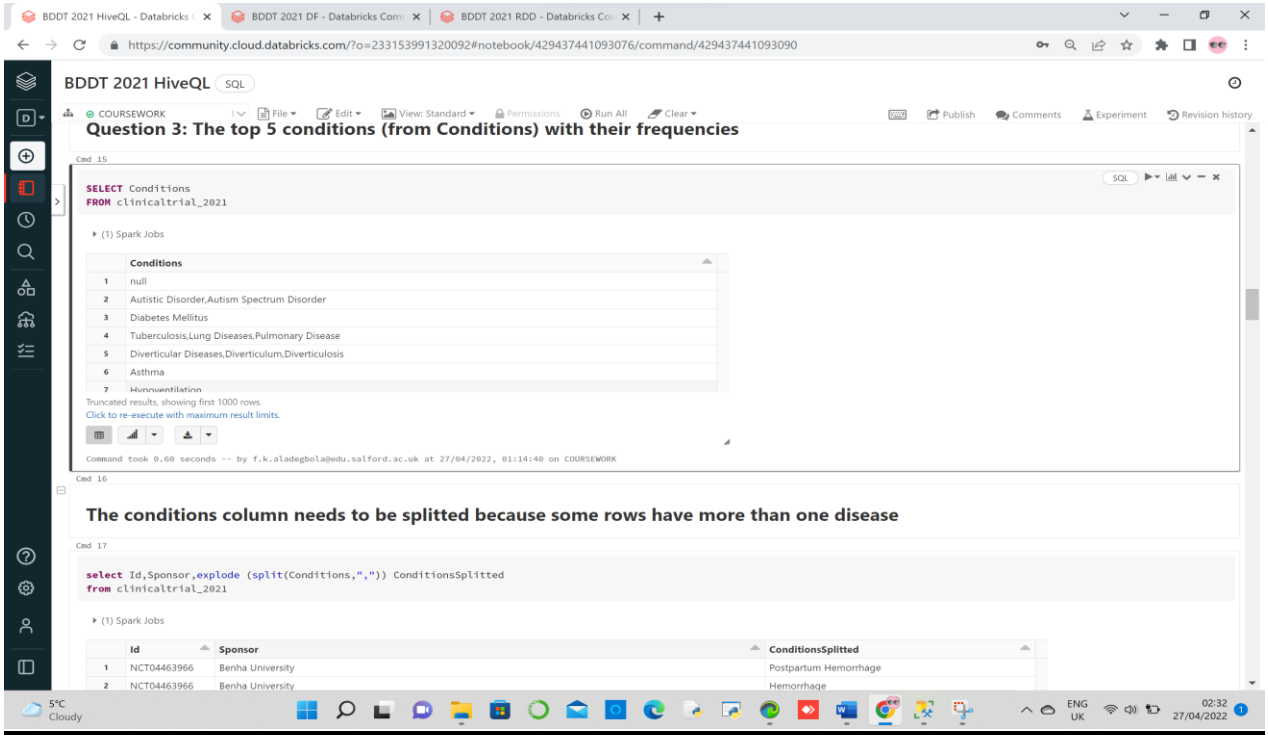
All the libraries connoted by (*) were imported from pyspark .sql functions. The 8th column i.e x[7] was passed through the anonymous function lambda to map the conditions column into a list. The flatMap function was adopted thereafter to split /flatten the row because there were more than one disease

on some of the rows. The reduceByKey was used afterward to list and aggregate the frequency of each condition.

- 

(3c.) **<u>HiveQL implementation outline</u>**

The function (explode(split,x)’,’)was adopted with the delimiter set as a comma in order to split the conditions column. The reason for the split stems from the fact that some of the rows in the conditions column have more than one condition per row.



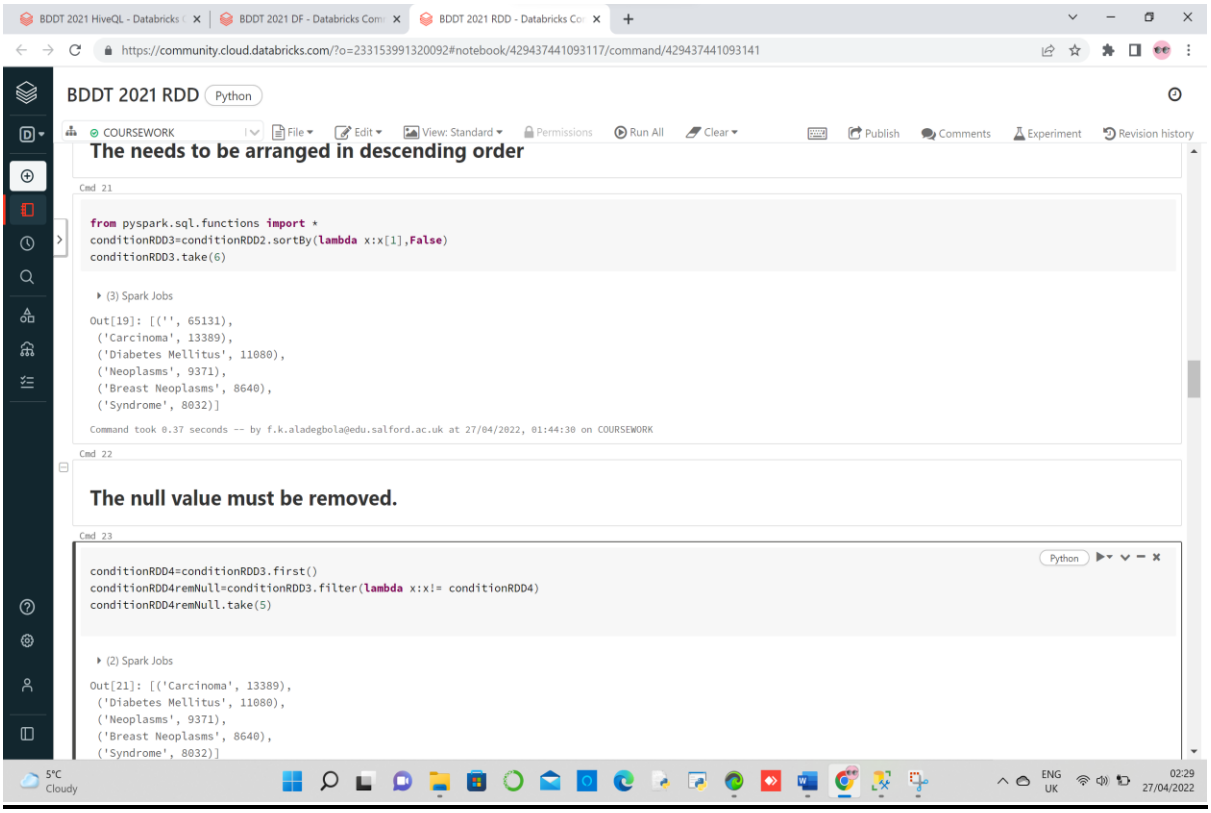
(3c.) **PySpark implementation outline DF**

Just like Hiveql the same function (explode(split,x)',') was used to split the rows with more than one condition per row. A new name with an alias 'conditions' was created.

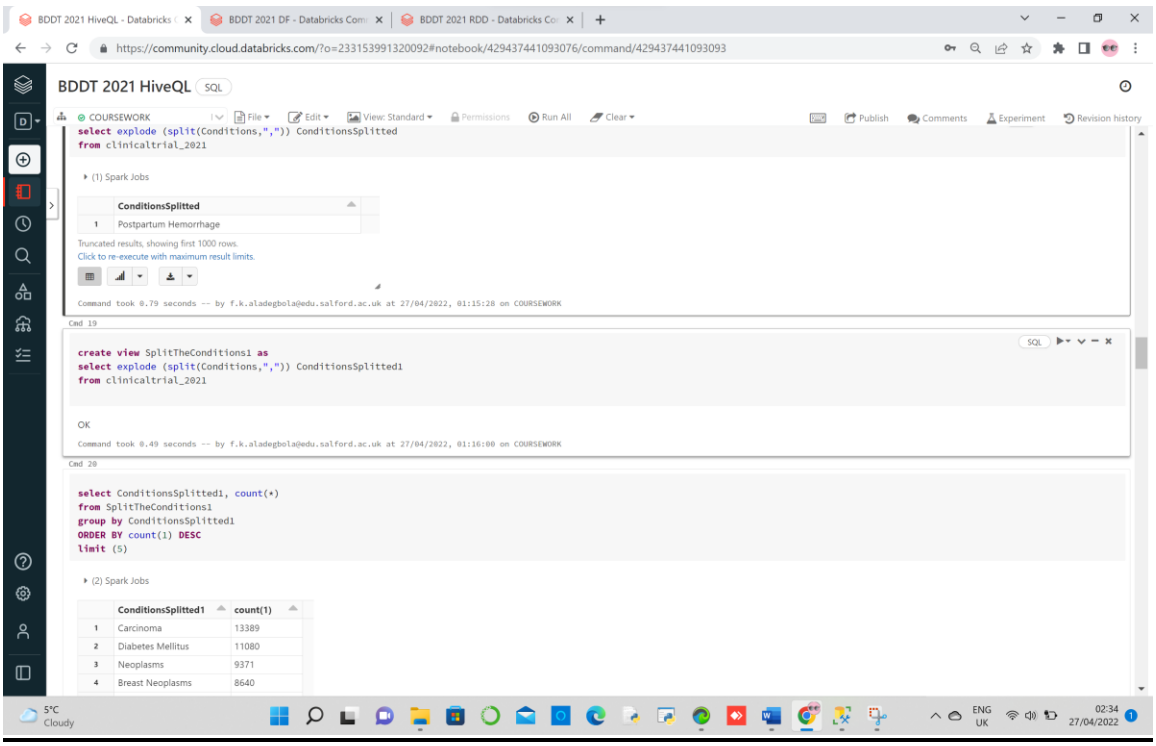
(3c.) **Result on the submission (final) data**

The arrangement in descending order for the three implementations follows the same process as the Type column.

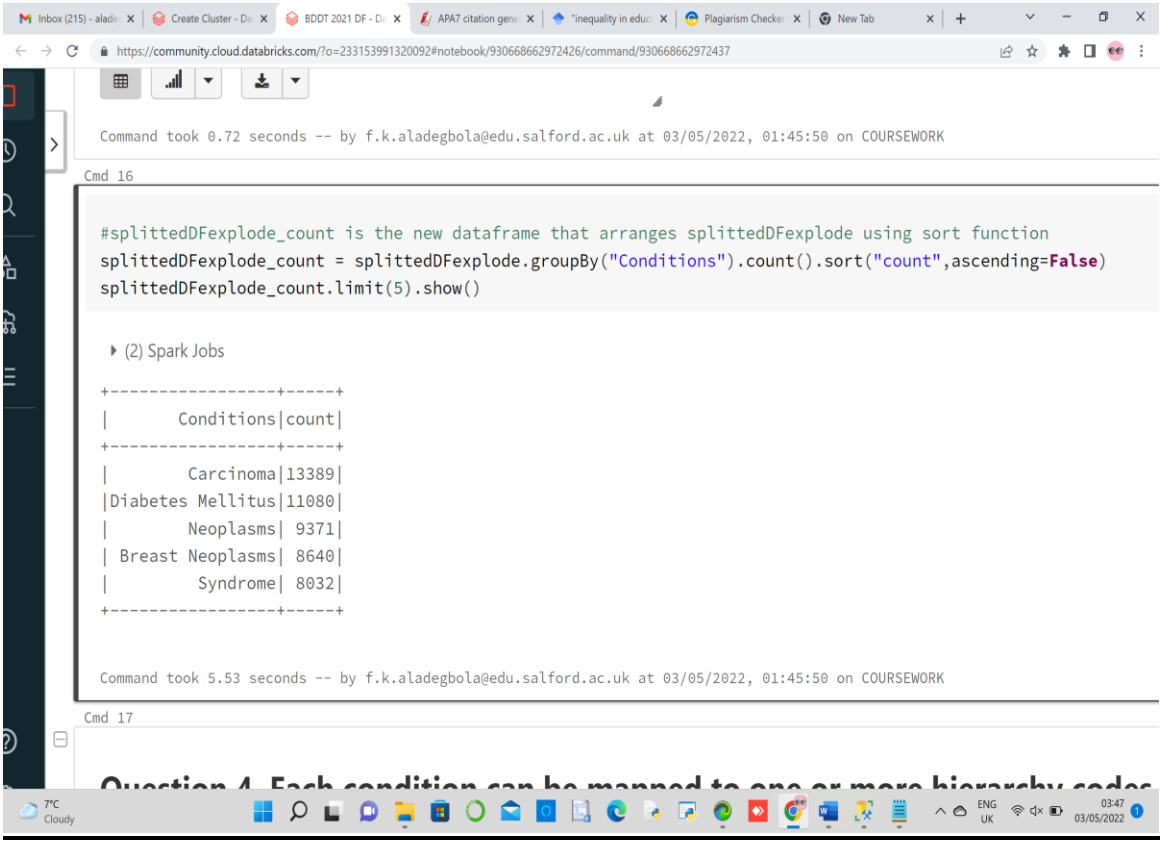
(3c.) **RDD:** Makes use of sortBy and the Boolean expression False for a descending order arrangement.



(3c.) **HIVE:** Makes use of order by with desc .



(3c.) **DATAFRAME:** Makes use of sort with Boolean expression False for a descending order arrangement.



(3c.) **Discussion of result:**

The three implementations gave the same result with carcinoma emerging as the condition with the highest frequency.

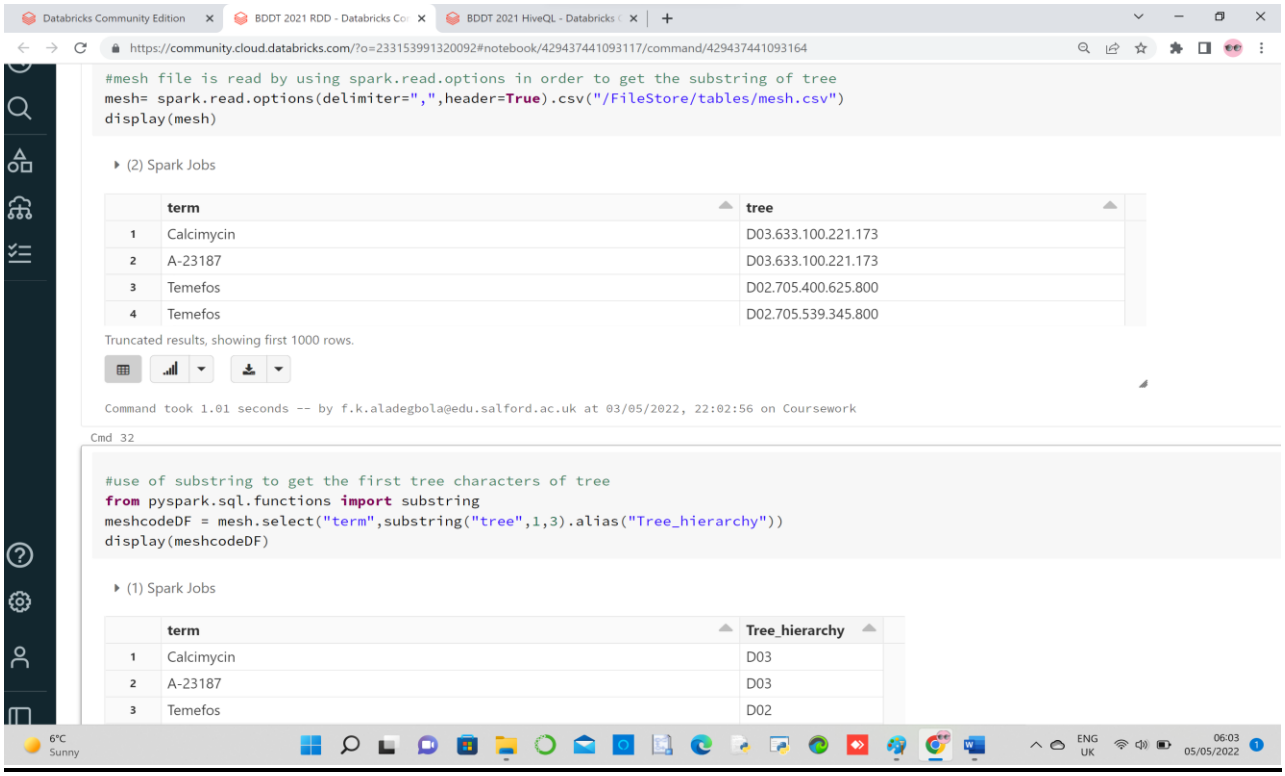
(3d.) **PROBLEM ANSWERS**

- **Question:** Each condition can be mapped to one or more hierarchy codes. The client wishes to know the 5 most frequent roots (i.e. the sequence of letters and numbers before the first full stop) after this is one.

  3d.) **Assumptions made:** It is assumed that the hierarchy codes are performed to categorize the medical terms reported approximately so that they can be analyzed or reviewed.

(3d.) **PySpark implementation outline RDD:**

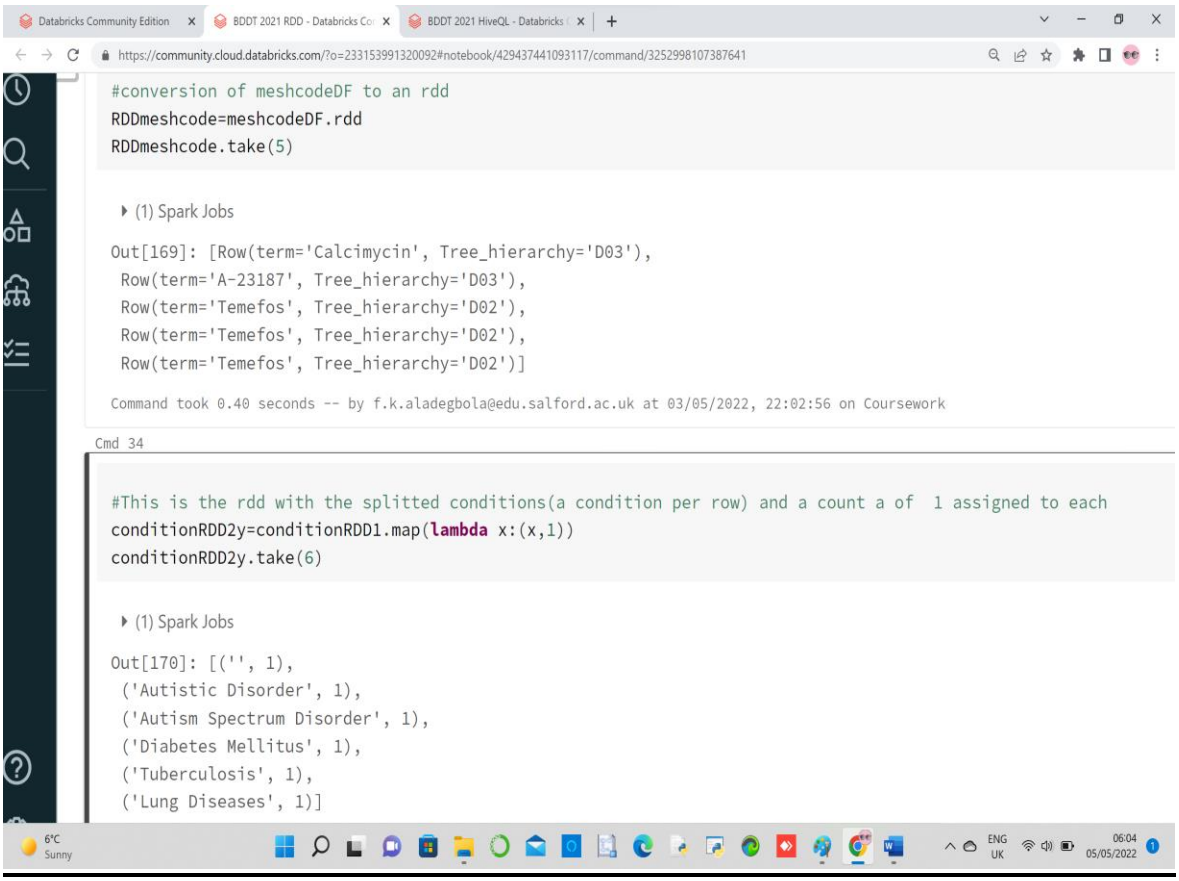
The mesh file is read through spark.read.options. This gives the mesh file in a tabular form. Then, the first three substrings of the column tree are selected using the statement 'substring' with an alias of Tree_Hierarchy. The selected term and substring tree are mapped to meshcodeDF



meshcodeDF is changed to an rdd using the dotrdd i.e (.rdd)

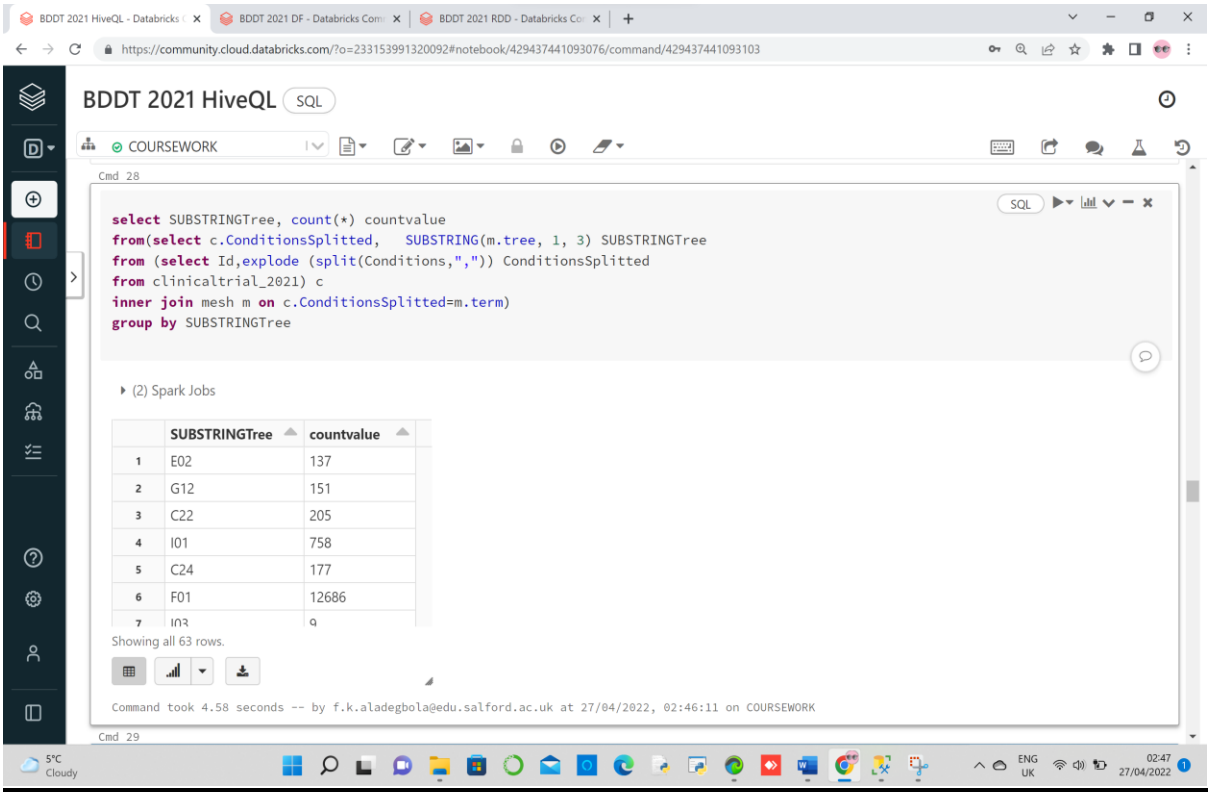
The rdd with the splitted conditions conditionRDD1, earlier in question 3

is mapped to a new rdd called conditionRDD2y by assigning a count of 1 to each condition.



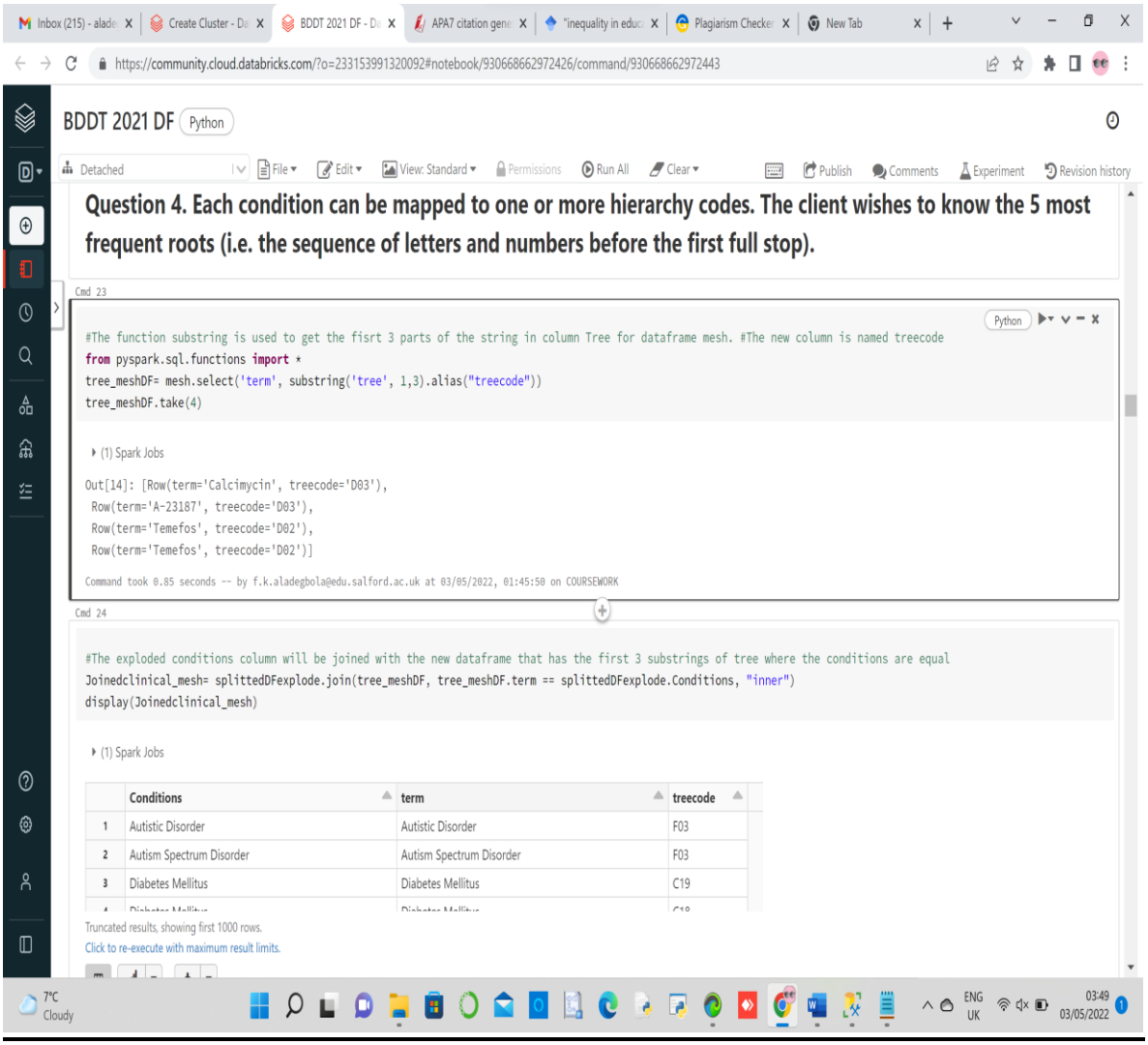
(3d.) **Hive implementation outline**

Hive makes use of select, inner join, and group by to map the first three substrings of Tree to the conditions in the clinical trial, with a count of it.



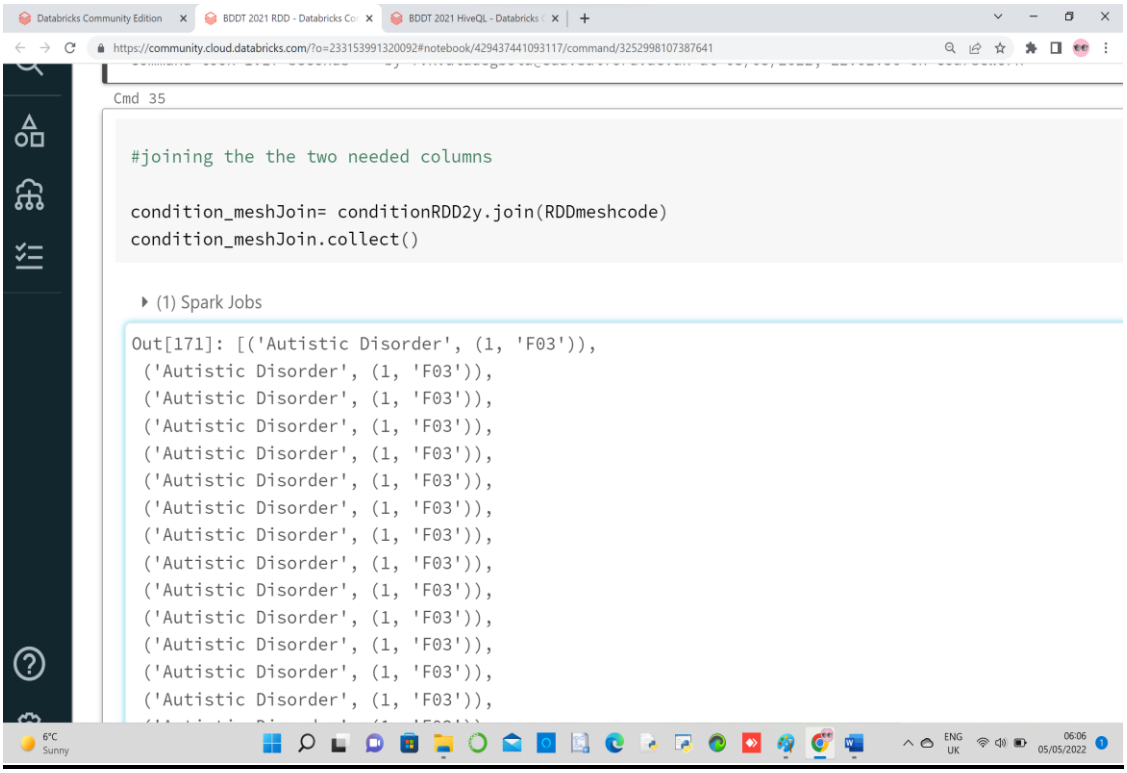
(3d.) **PySpark implementation outline DF**

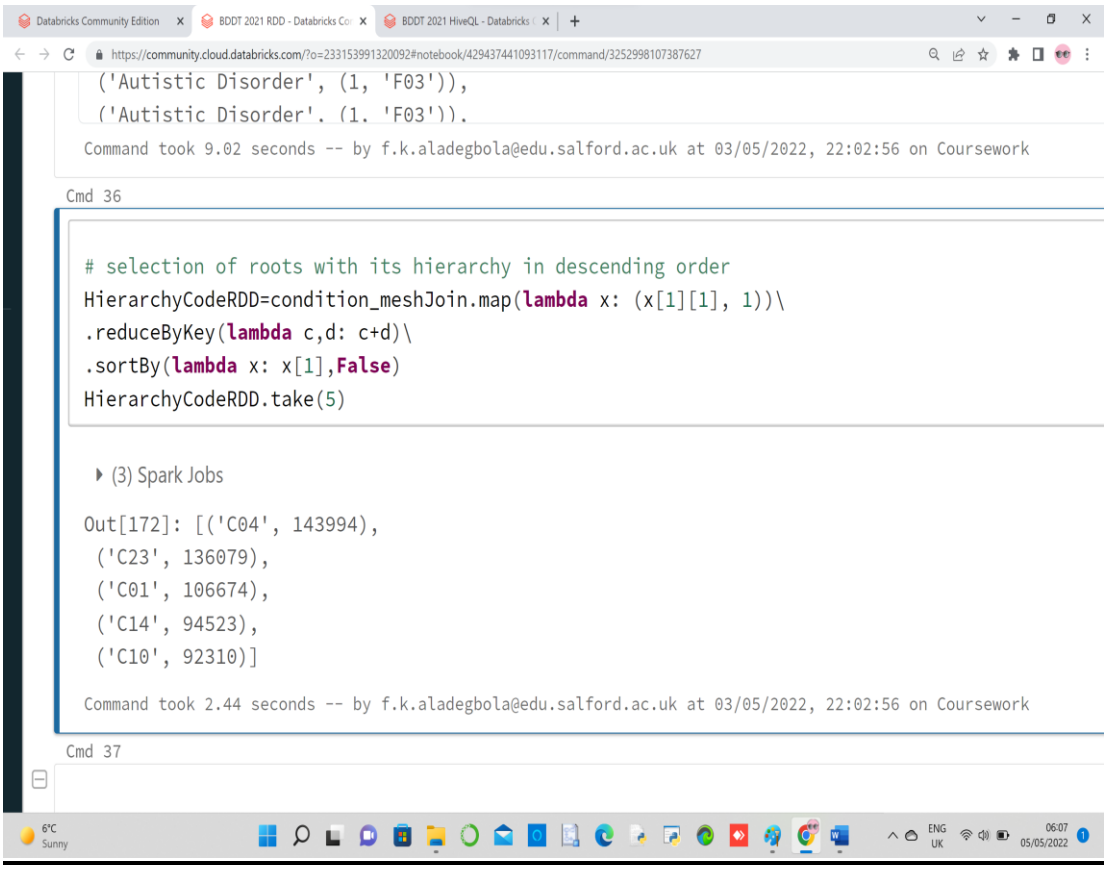
All libraries are imported from pyspark.sql.functions with *, then substring is used to select the term column and the first three substrings of the tree column. The resulting rdd tree_meshDF is joined to the exploded rdd in question 3, splittedDFexplode. This creates a table with the columns conditions, term and the first three substrings of tree.



(3d.) **Result on the submission (final) data**

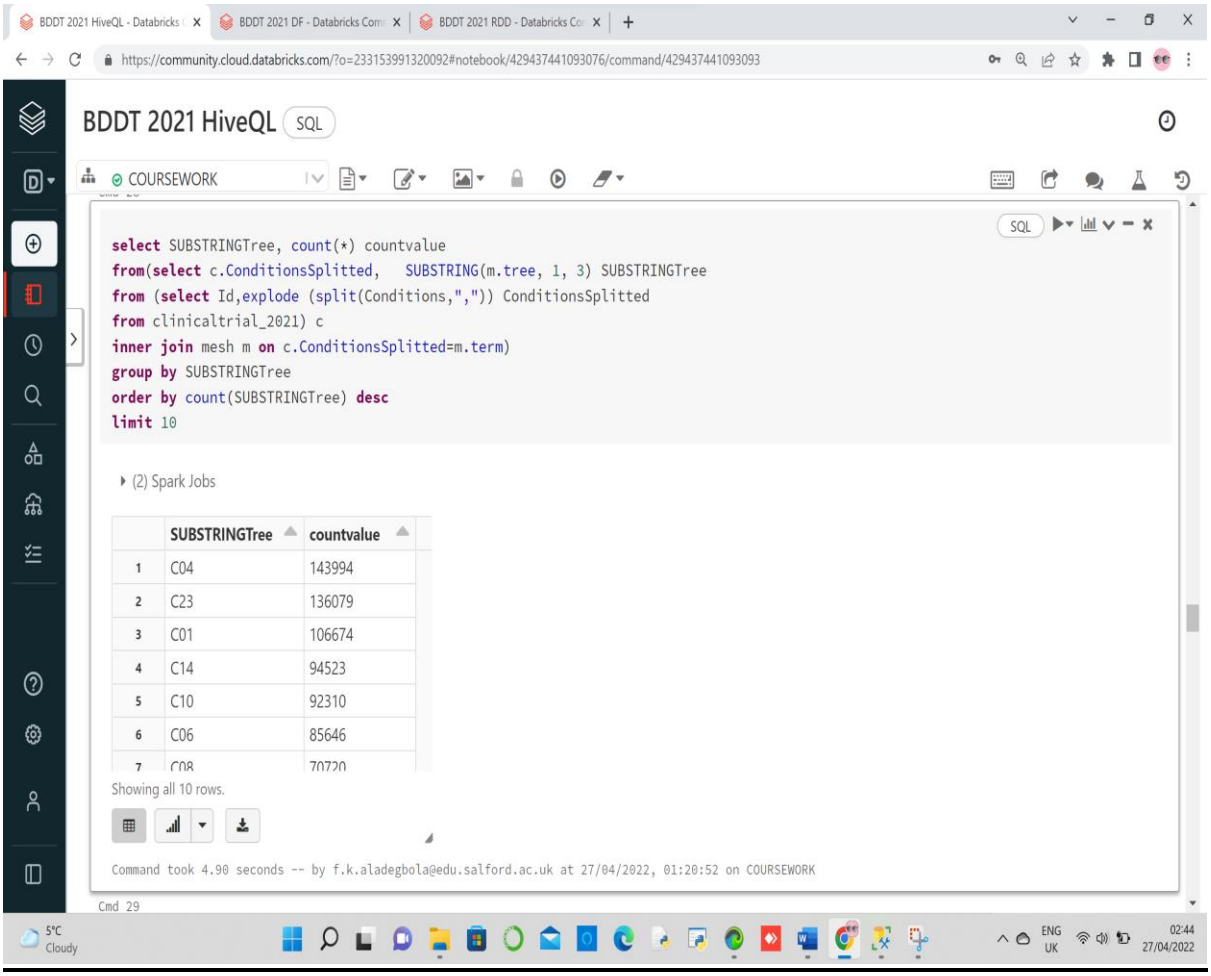
(3d.) **RDD:** The result for question 4 under rdd is mapped to HierarchycodeRDD with the use of map, lambda, reduceBKey, sortBy, and the Boolean expression False to obtain a count, get a frequency and arrange in descending order.


Databricks Community Edition
BDDT 2021 RDD - Databricks
BDDT 2021 HiveQL - Databricks
https://community.cloud.databricks.com/?o=2331539913200928notebook/4294374410931178command/3252998107387641
Cmd 35
#joining the the two needed columns
condition_meshJoin= conditionRDD2y.join(RDDmeshcode)
condition_meshJoin.collect()
(1) Spark Jobs
Out[171]: [('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
6°C
Sunny
ENG
UK
06:06
05/05/2022


Databricks Community Edition
BDDT 2021 RDD - Databricks
BDDT 2021 HiveQL - Databricks
https://community.cloud.databricks.com/?o=2331539913200928notebook/4294374410931178command/3252998107387627
('Autistic Disorder', (1, 'F03')),
('Autistic Disorder', (1, 'F03')),
Command took 9.02 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 22:02:56 on Coursework
Cmd 36
# selection of roots with its hierarchy in descending order
HierarchyCodeRDD=condition_meshJoin.map(lambda x: (x[1][1], 1))\
.reduceByKey(lambda c,d: c+d)\
.sortBy(lambda x: x[1],False)
HierarchyCodeRDD.take(5)
(3) Spark Jobs
Out[172]: [('C04', 143994),
('C23', 136079),
('C01', 106674),
('C14', 94523),
('C10', 92310)]
Command took 2.44 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 22:02:56 on Coursework
Cmd 37
6°C
Sunny
ENG
UK
06:07
05/05/2022

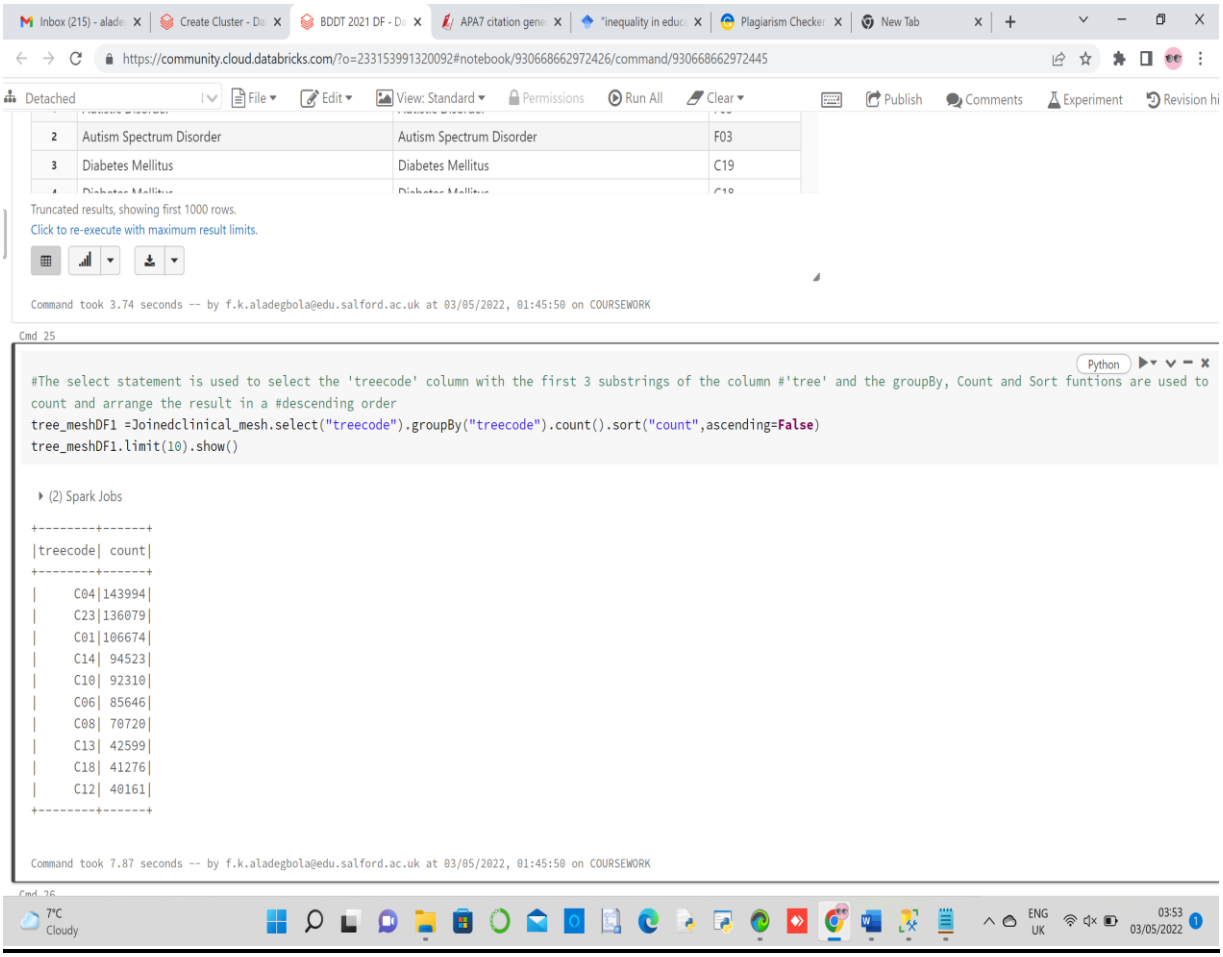
(3d.) **HIVE**

The select statement is used to pick the relevant columns with an inner join. The group by, order by and limit are finally used to get a frequency of the first three substring of the column tree arranged in a descending order Frquency.



(3d.) **Data frame**

The final result is mapped in a dataframe called tree_meshDF1.it includes the joined tables. The statement ‘grouby’ and ‘sort’ with boolean expression False are used to get a frequency of the hierarchy code in descending order.



(3d.) **Discussion of result:**

The three implementations produced the same result with the top hierarchies having C as their first substring.

(3e.)**PROBLEM ANSWERS**

Question:5

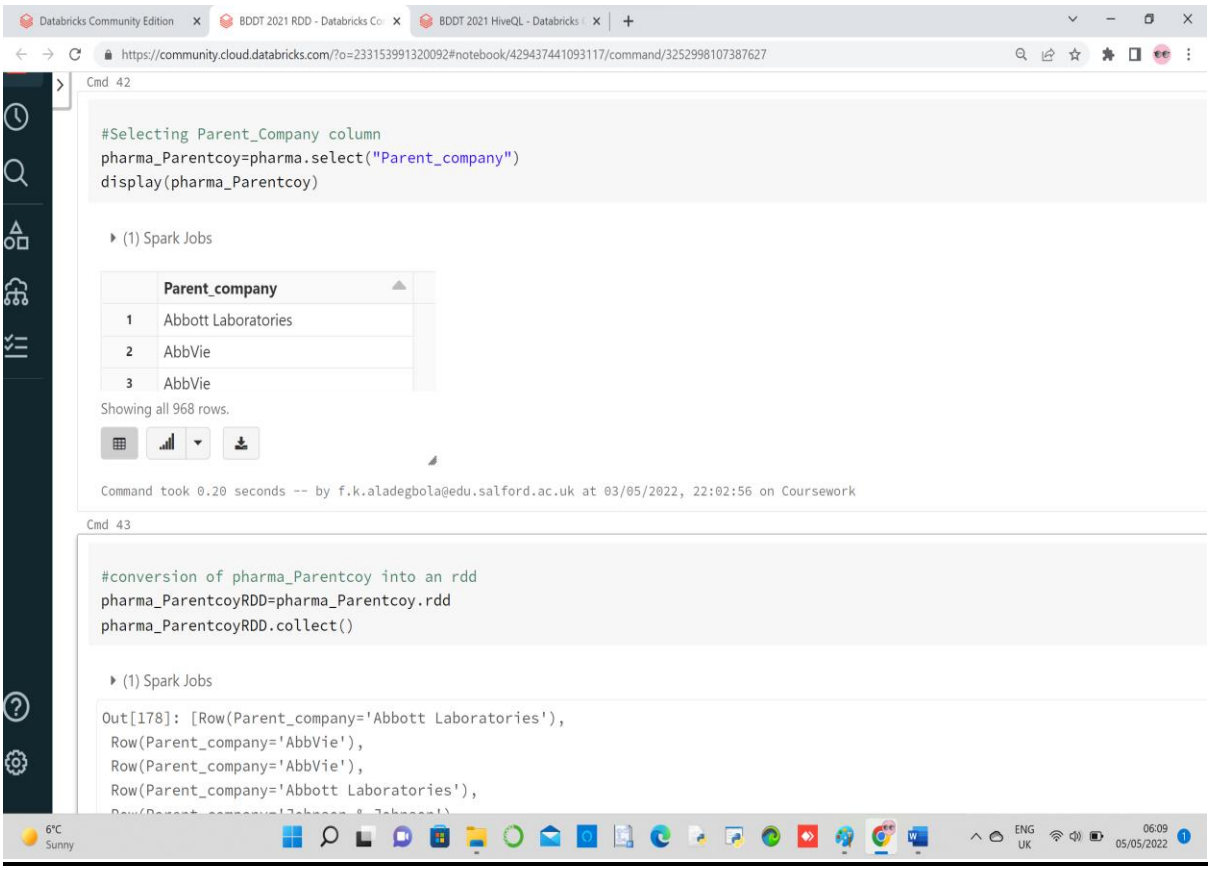
Find the 10 most common sponsors that are not pharmaceutical companies, along with the number of clinical trials they have sponsored.

(3e.)**Assumptions made:**

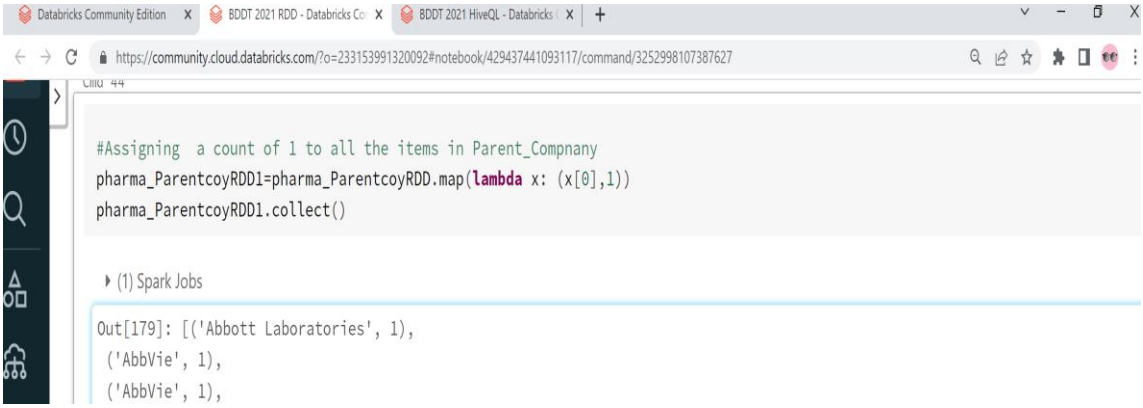
It was assumed that the Parent Company column in the pharma file contains all possible pharmaceutical companies.

(3e.)**PySpark implementation outline RDD**

The Column Parent_company was selected from pharma file to form pharma_Parentcoy. This was converted to an rdd called pharma_ParentcoyRDD using the command (.rdd).

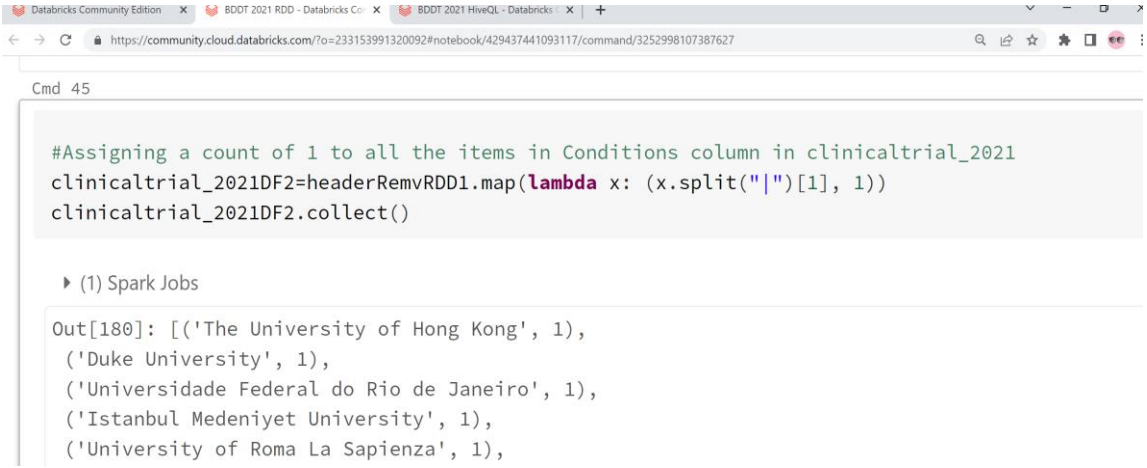


The number 1 was assigned to each row in pharma_ParentcoyRDD using map and lambda to get a count.



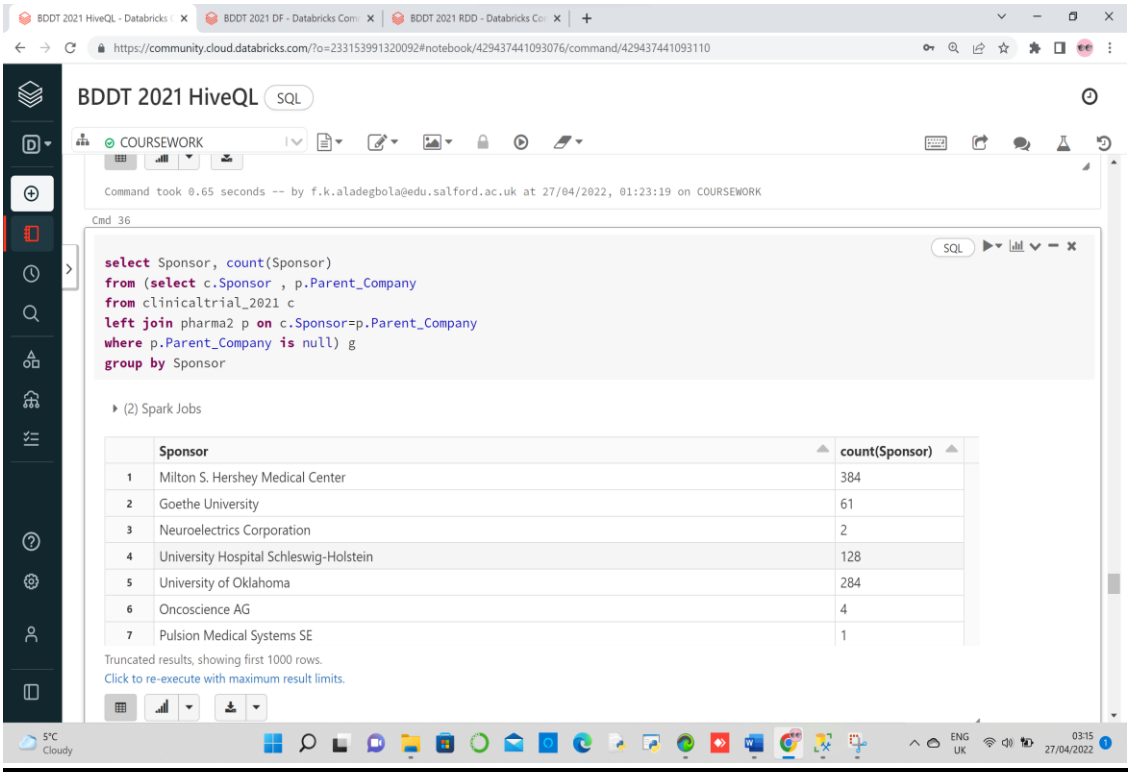
headerRemvRDD1 is clinicaltrial_2021 file with the header removed.

The number 1 was assigned to each row in headerRemvRDD1 using map, lambda and split to another rdd called clinicaltrial_2021DF2. The essence of this is to get an individual count of the rows in the column sponsors.



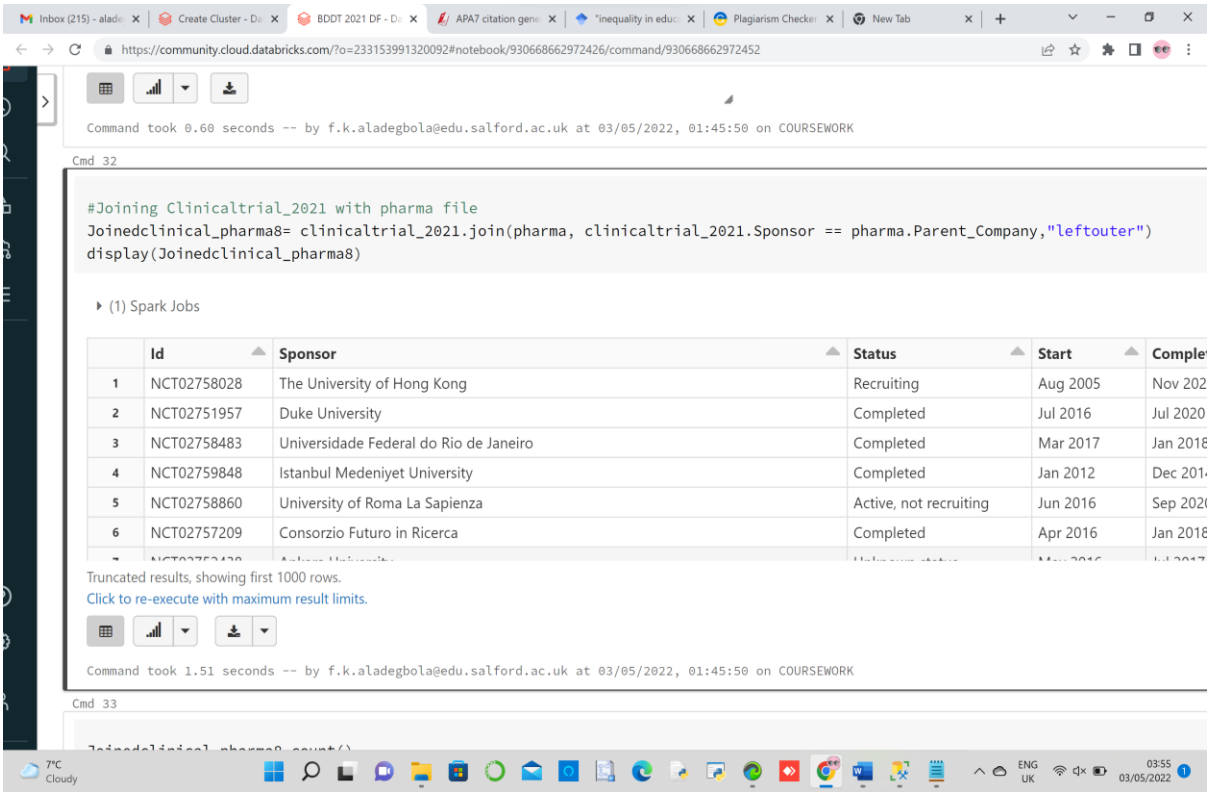
(3e.) **HiveQL implementation outline**

In Hive, the statements ‘select’, ‘left join’ and ‘group by’ were used to get a table with the sponsors that are not pharmaceuticals. This was achieved with a left join of the sponsors in clinicaltrial_2021 and Parent_company in Pharma file where the two columns are equal.



3e.)**Dataframe implementation outline**

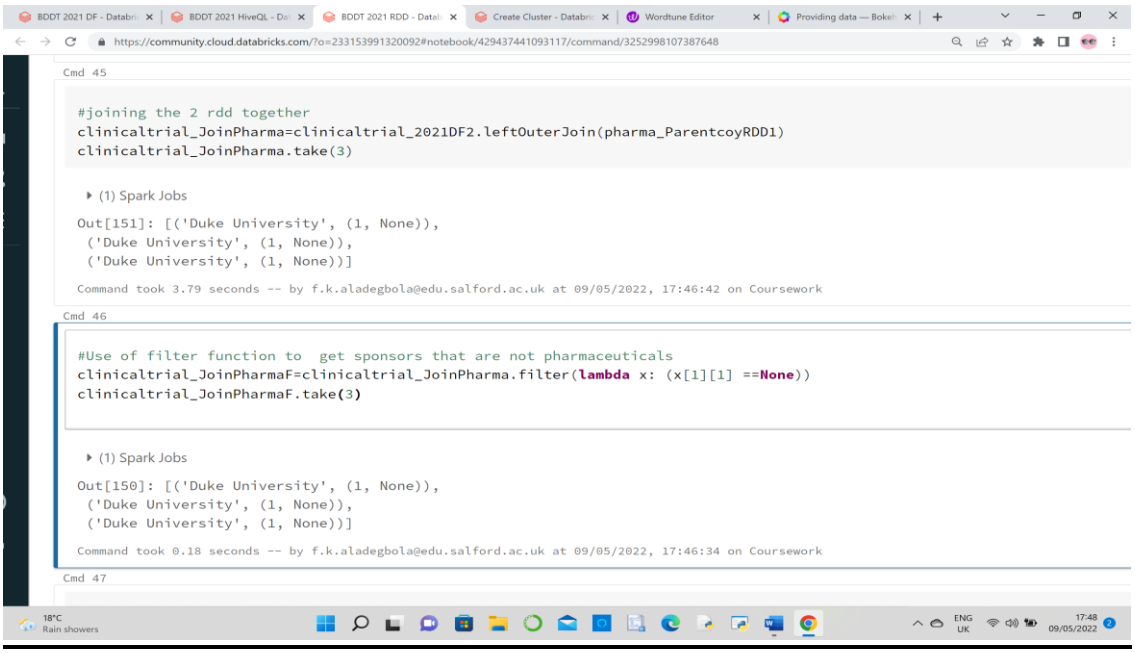
The function left outer join was used to join clinicaltrial_2021 with pharma file where sponsor and parent_Company were equal.



(3e.)**Result on the submission (final) data –**

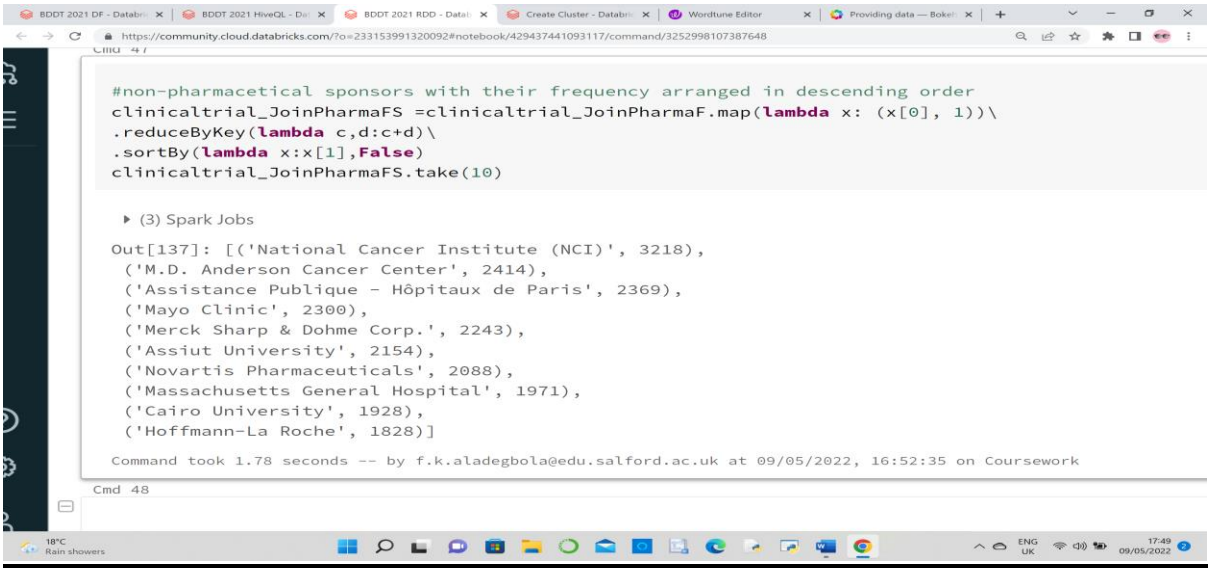
(3e.)**RDD**

Clinicaltrial_JoinPharmaF was used as the rdd where clinicaltrial_JoinPharma as filtered for null values. The null values are the sponsors that are not pharmaceuticals.



```
Cmd 45

#joining the 2 rdd together
clinicaltrial_JoinPharma=clinicaltrial_2021DF2.leftOuterJoin(pharma_ParentcoyRDD1)
clinicaltrial_JoinPharma.take(3)

▸ (1) Spark Jobs

Out[151]: [('Duke University', (1, None)),
 ('Duke University', (1, None)),
 ('Duke University', (1, None))]

Command took 3.79 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 09/05/2022, 17:46:42 on Coursework

Cmd 46

#Use of filter function to  get sponsors that are not pharmaceuticals
clinicaltrial_JoinPharmaF=clinicaltrial_JoinPharma.filter(lambda x: (x[1][1] ==None))
clinicaltrial_JoinPharmaF.take(3)

▸ (1) Spark Jobs

Out[150]: [('Duke University', (1, None)),
 ('Duke University', (1, None)),
 ('Duke University', (1, None))]

Command took 0.18 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 09/05/2022, 17:46:34 on Coursework

Cmd 47
```

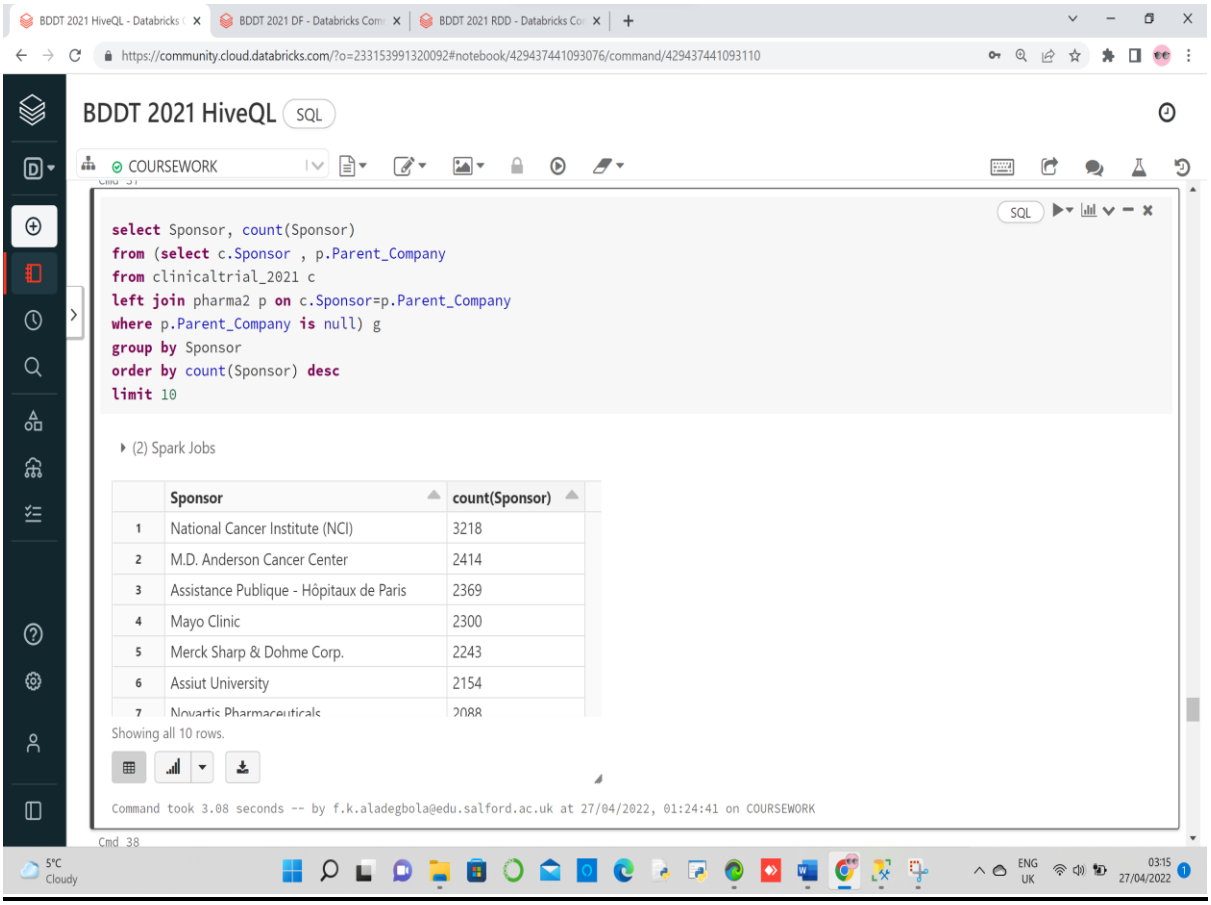
The reduceByKey, sortBy, and Boolean expression False was used to arrange the result in a descending order.



```
#non-pharmacetical sponsors with their frequency arranged in descending order
clinicaltrial_JoinPharmaFS =clinicaltrial_JoinPharmaF.map(lambda x: (x[0], 1))\
.reduceByKey(lambda c,d:c+d)\
.sortBy(lambda x:x[1],False)
clinicaltrial_JoinPharmaFS.take(10)

▸ (3) Spark Jobs

Out[137]: [('National Cancer Institute (NCI)', 3218),
 ('M.D. Anderson Cancer Center', 2414),
 ('Assistance Publique - Hôpitaux de Paris', 2369),
 ('Mayo Clinic', 2300),
 ('Merck Sharp & Dohme Corp.', 2243),
 ('Assiut University', 2154),
 ('Novartis Pharmaceuticals', 2088),
 ('Massachusetts General Hospital', 1971),
 ('Cairo University', 1928),
 ('Hoffmann-La Roche', 1828)]

Command took 1.78 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 09/05/2022, 16:52:35 on Coursework

Cmd 48
```

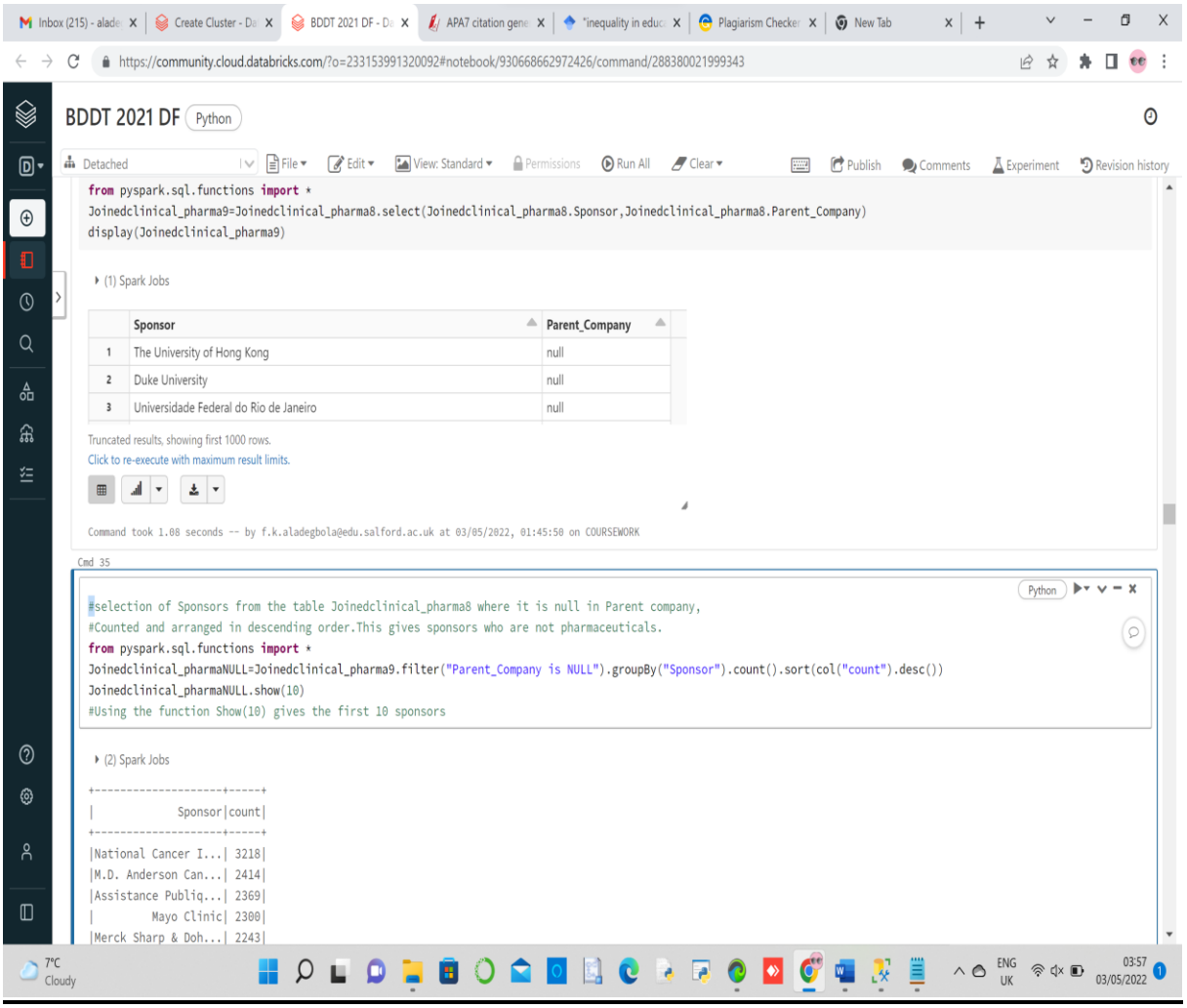
(3e.)**HIVE**

group by, order by and limit were used to arrange the result in descending order.



(3e.)**DATAFRAME**

In Dataframe implementation, the functions filter where Parent_company was null was used, with groupBy and sort to get a frequency in descending order of sponsors that are not pharmaceuticals.



(3e.)**Discussion of result:**

National Cancer Institute (NCI) emerged as the non-pharmaceutical sponsor with the highest number of clinical trials. This was clearly shown by the result of the 3 implementations. This shows that quite a lot of the clinical trials were cancer-related and the sponsors are keen on sponsoring clinical trials to get results for drugs to be used.

(3f.) **PROBLEM ANSWERS**

Question:6

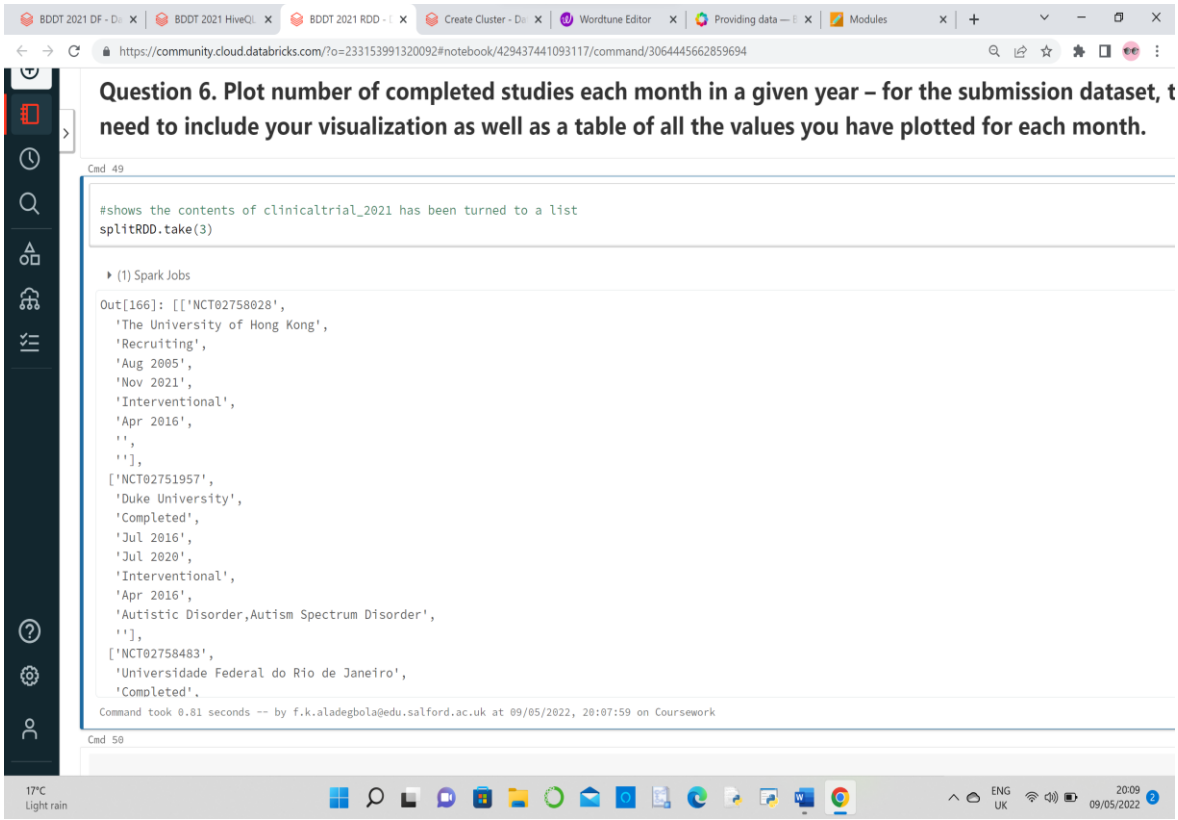
Plot the number of completed studies each month in a given year – for the submission dataset, the year is 2021. You need to include your visualization as well as a table of all the values you have plotted for each month.

(3f.) **Assumptions made:**

It was assumed that the completed studies for each year were properly recorded and there was no error in documentation

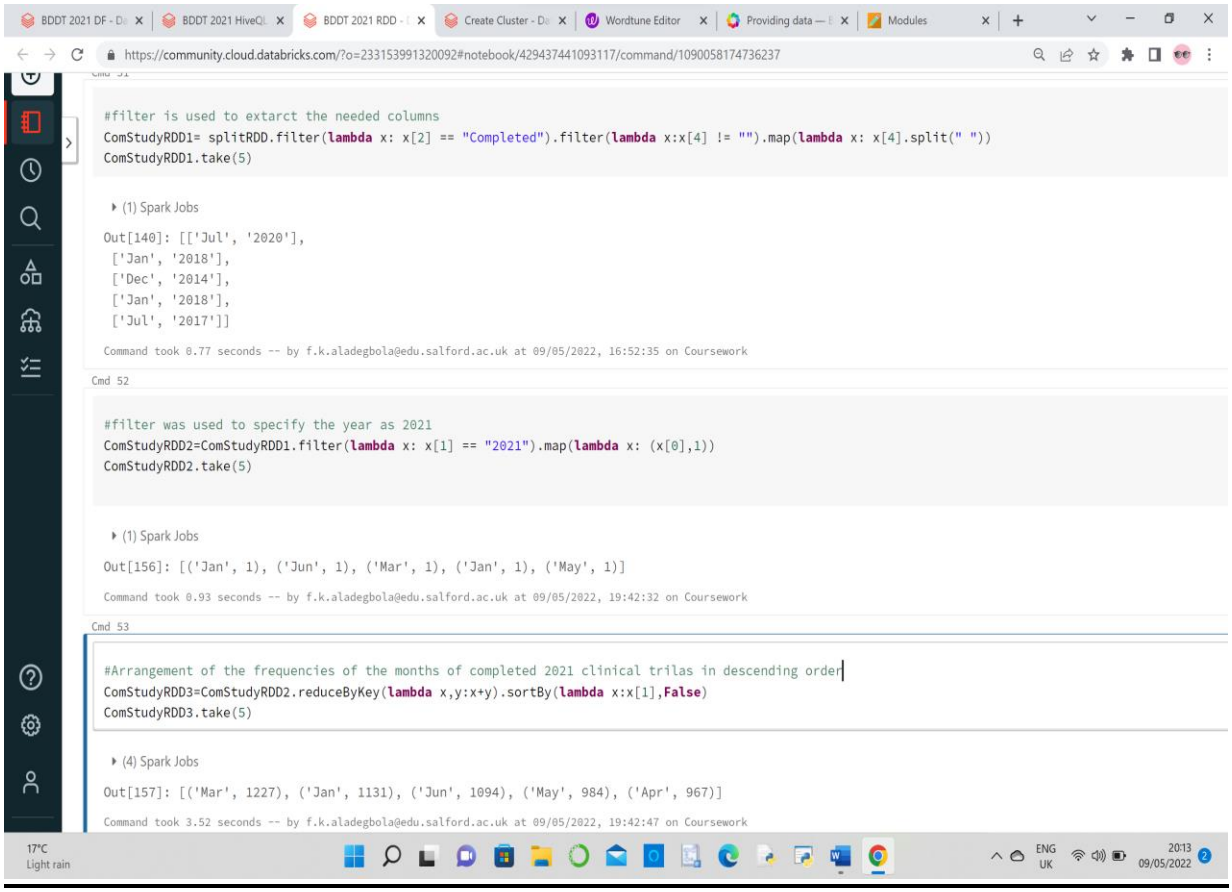
(3f.) **PySpark implementation outline_RDD**

Using take() on splitRDD the rdd from clinicaltrial_2021 turned to a list, gives a view of the items in the data according to columns. x[0] represents the Id, x[1]represents sponsors e.t.c



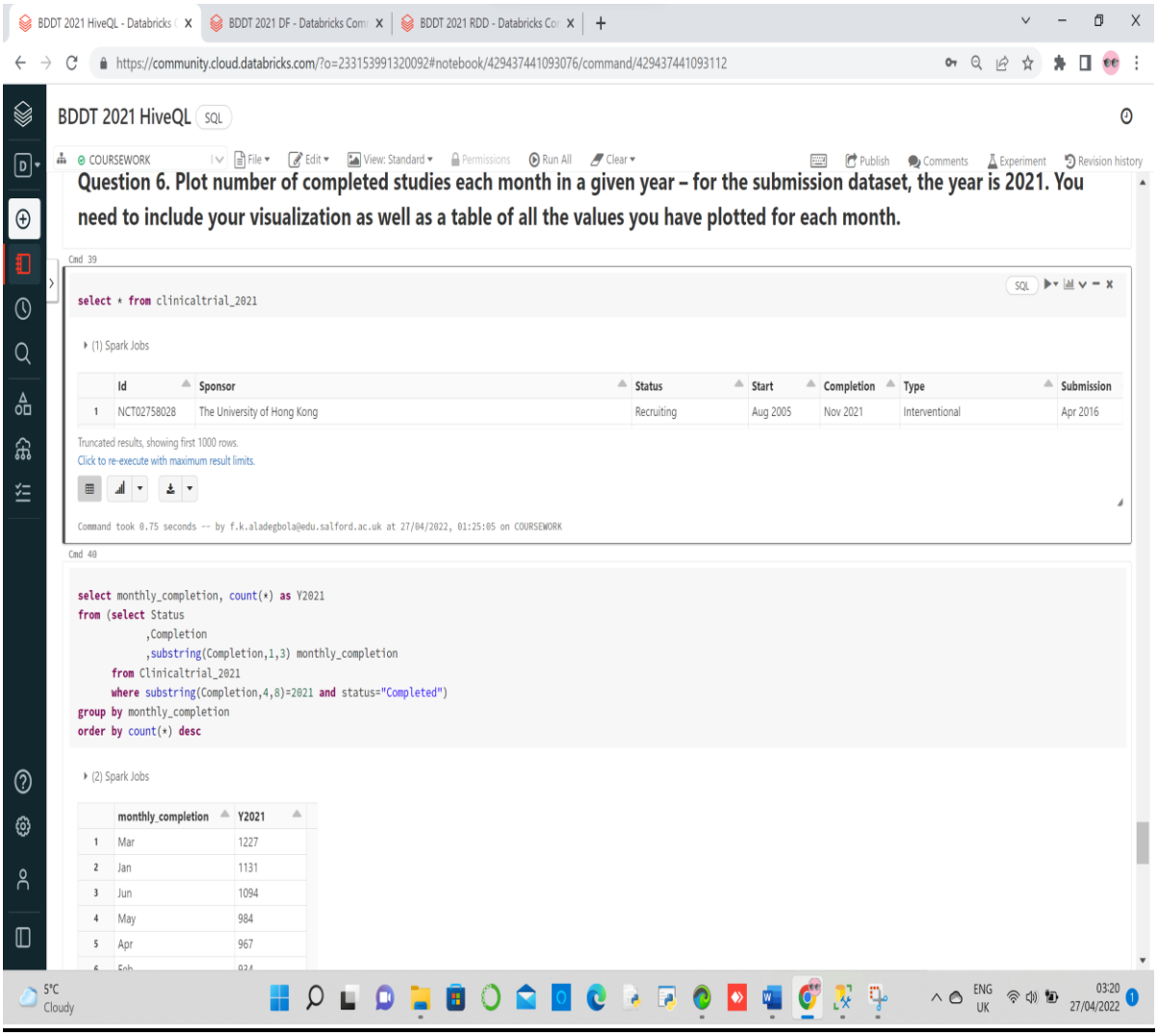
Filter function was used to extract the needed columns and also to specify the year of completed studies as 2021.

Thereafter, each month was given a count of 1, then reduceByKey and Sort was used to calculate the total frequency and to arrange the months of completed studies in descending order.



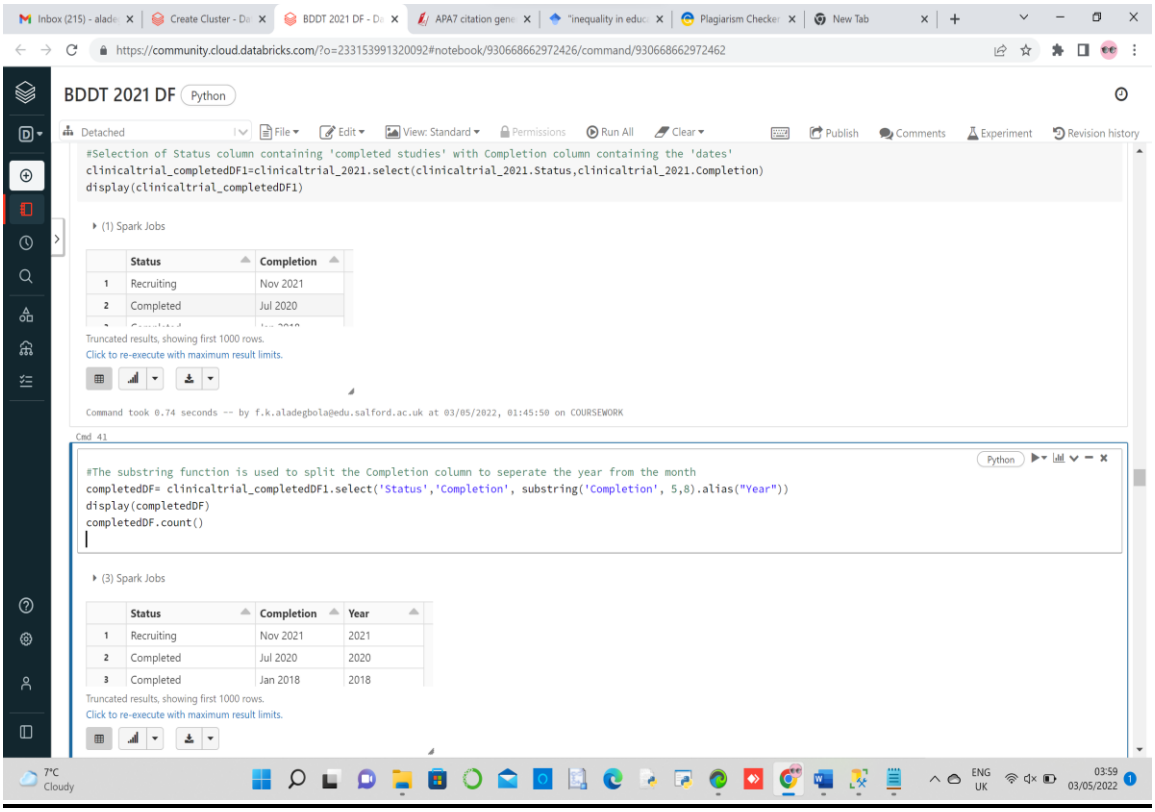
(3f.) **HiveQL implementation outline**

The select * statement meaning to select all was used to recall the data in clinicaltrial_2021. The function substring was used to get the months and the year and the ‘where’ clause specified the year as 2021.

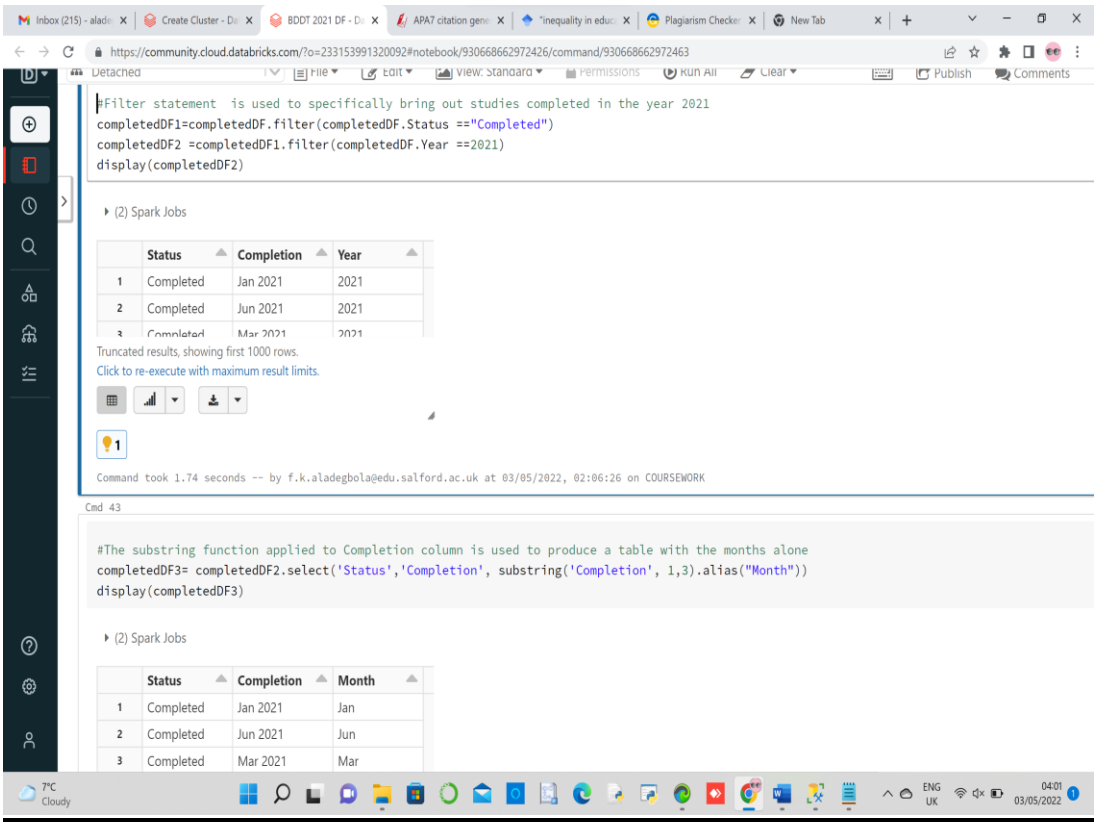


(3f.) **Dataframe implementation outline**

The select statement and substring were also used in dataframes to get a table for 2021 completion monthly. Substring(Completion 5,8) brings out the year.

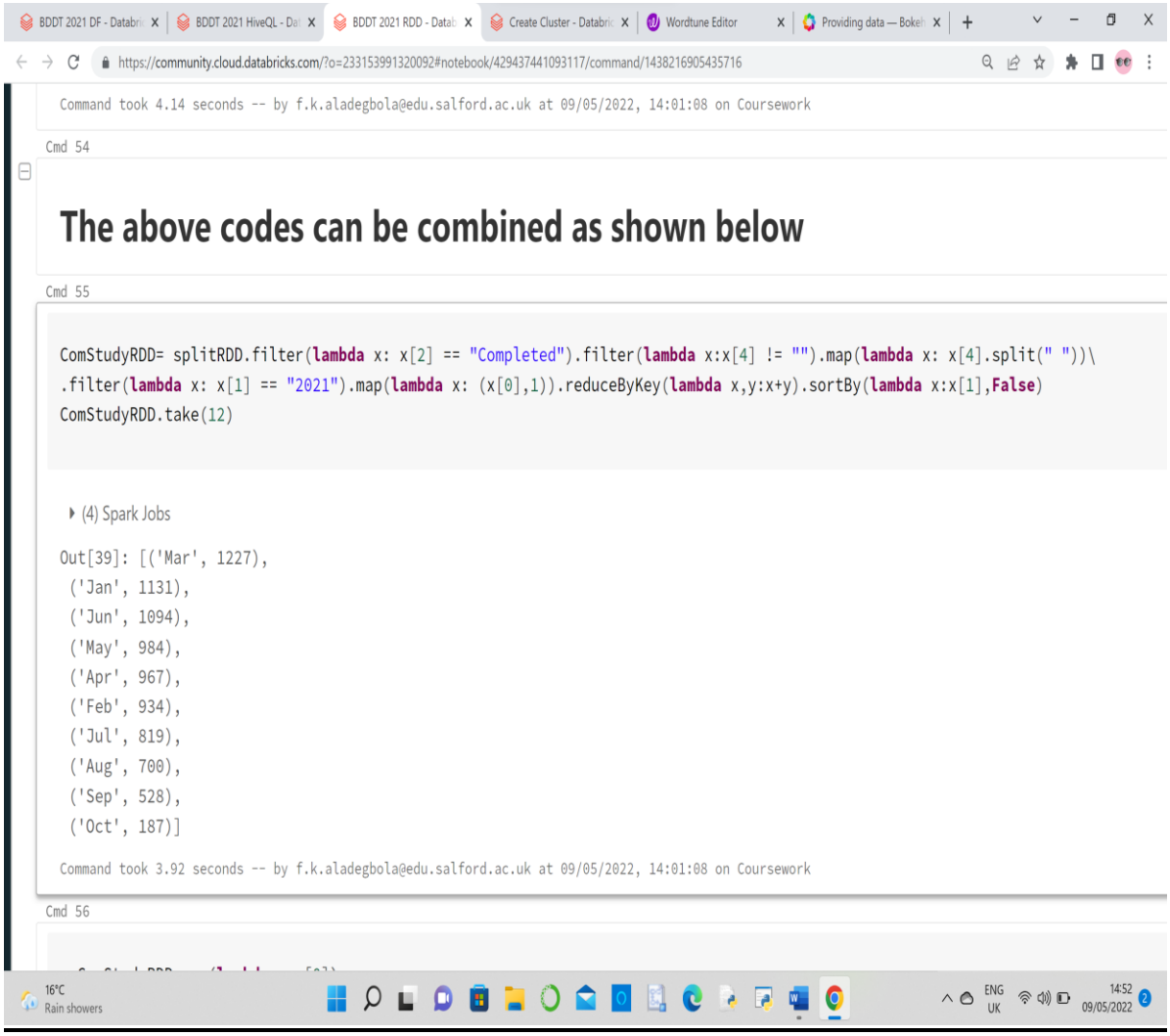


completedDF is filtered to specify status as completed only while completedDF1 is filtered for the specific year to be 2021.



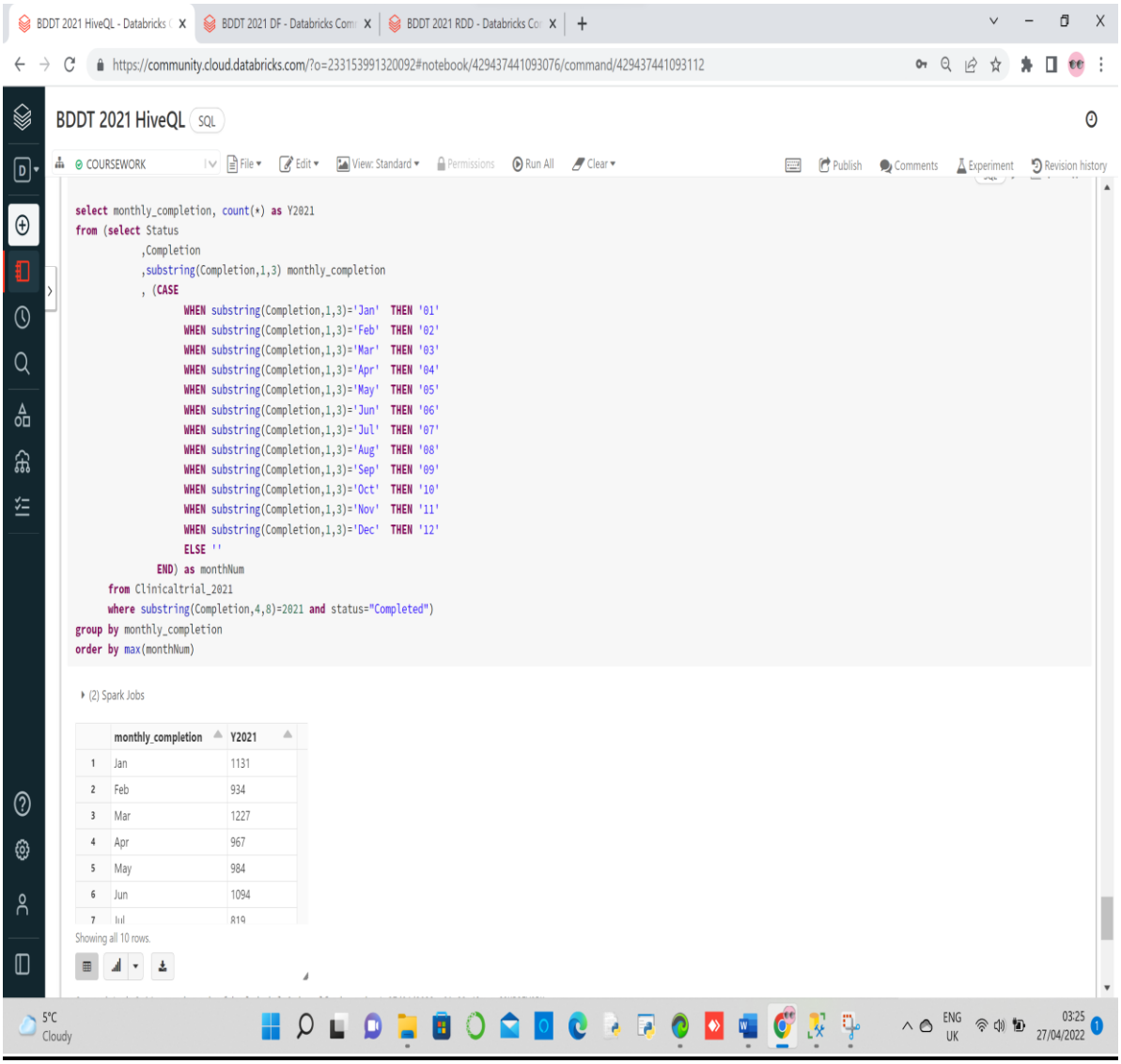
(3f.) **Result on the submission (final) data –**

(3f.) **RDD:** The reduceByKey and SortBy with the Boolean value False was used to arrange the months of completed study in 2021.

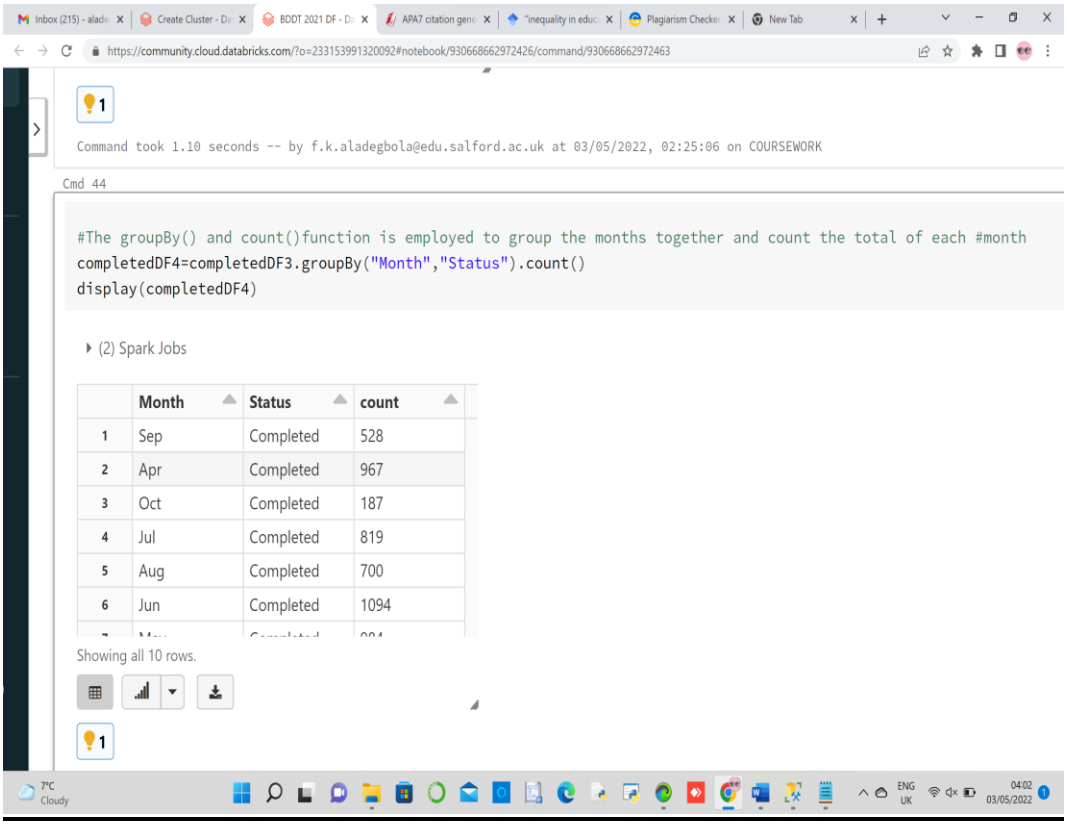


(3f.) **HIVE**

Count, substring group by and order by were used to arrange the months in the calendar arrangement.



(3f.) **DATA FRAME:** The use of groupBy and count was employed to get the calender months of completed study in 2021.



	Month	Status	count
1	Sep	Completed	528
2	Apr	Completed	967
3	Oct	Completed	187
4	Jul	Completed	819
5	Aug	Completed	700
6	Jun	Completed	1094

- **Discussion of result:**

	monthly_completion	Y2021
1	Mar	1227
2	Jan	1131
3	Jun	1094
4	May	984
5	Apr	967
6	Feb	934
7	Jul	819
8	Aug	700
9	Sep	528
10	Oct	187

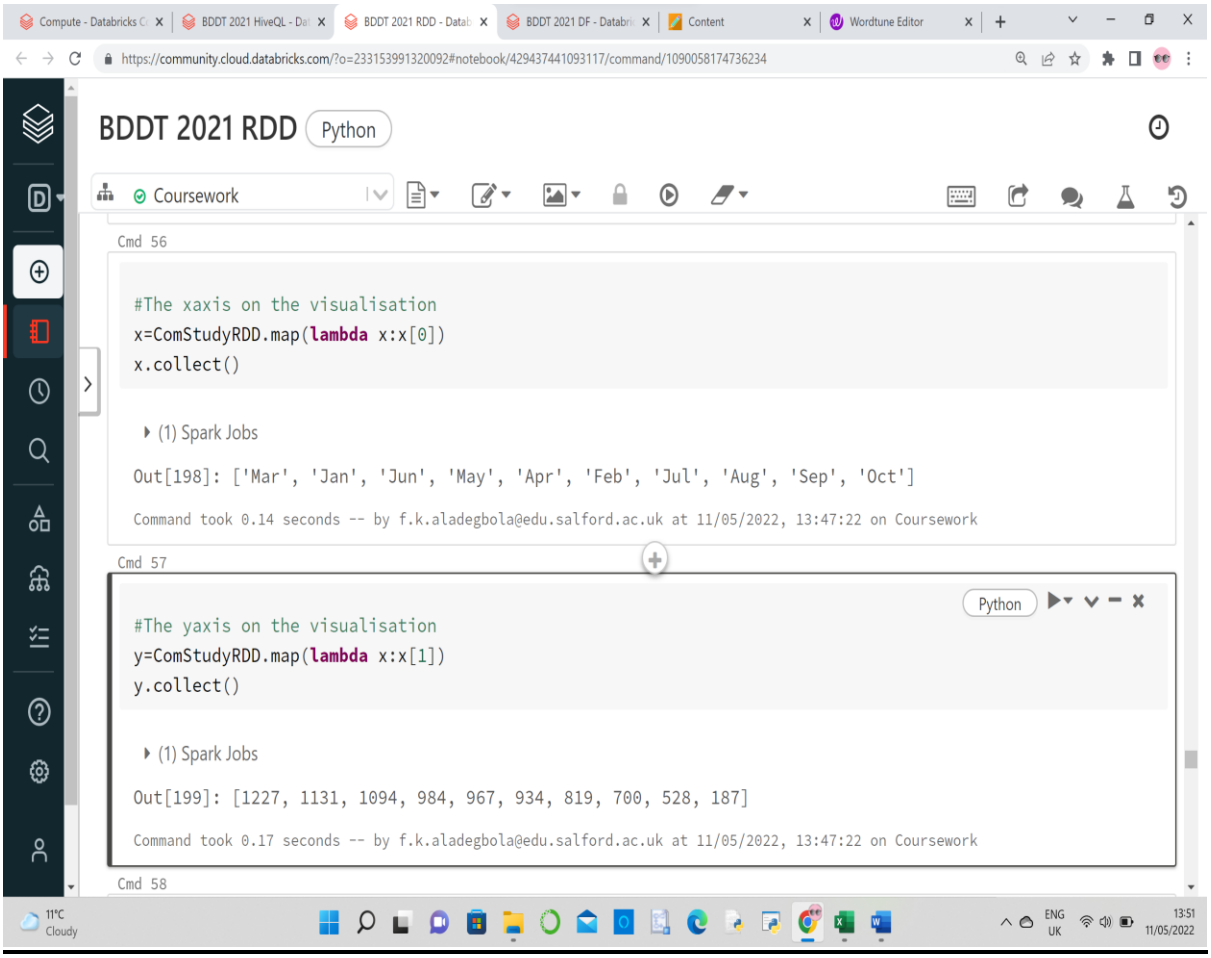
The highest number of completed studies took place in the month of March, followed by January and then June.

(3f.) **VISUALISATIONS**

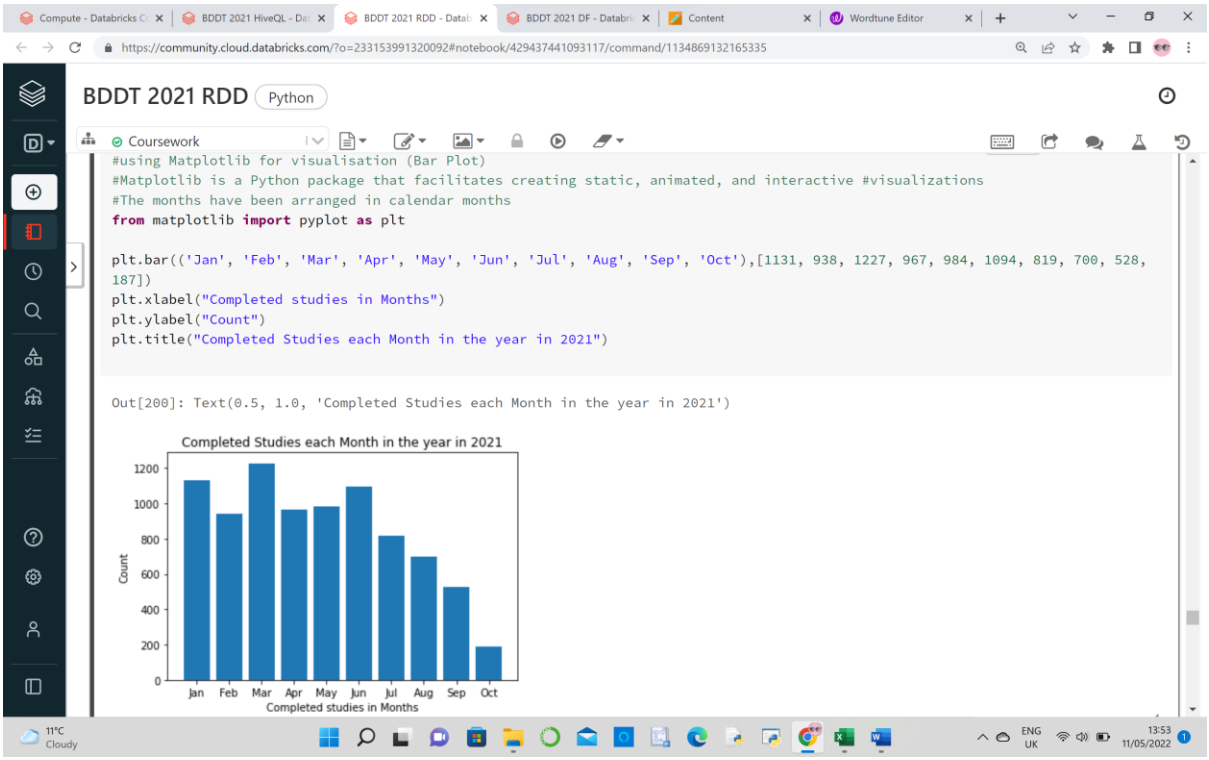
RDD

Matplotlib was imported for visualization in RDD.

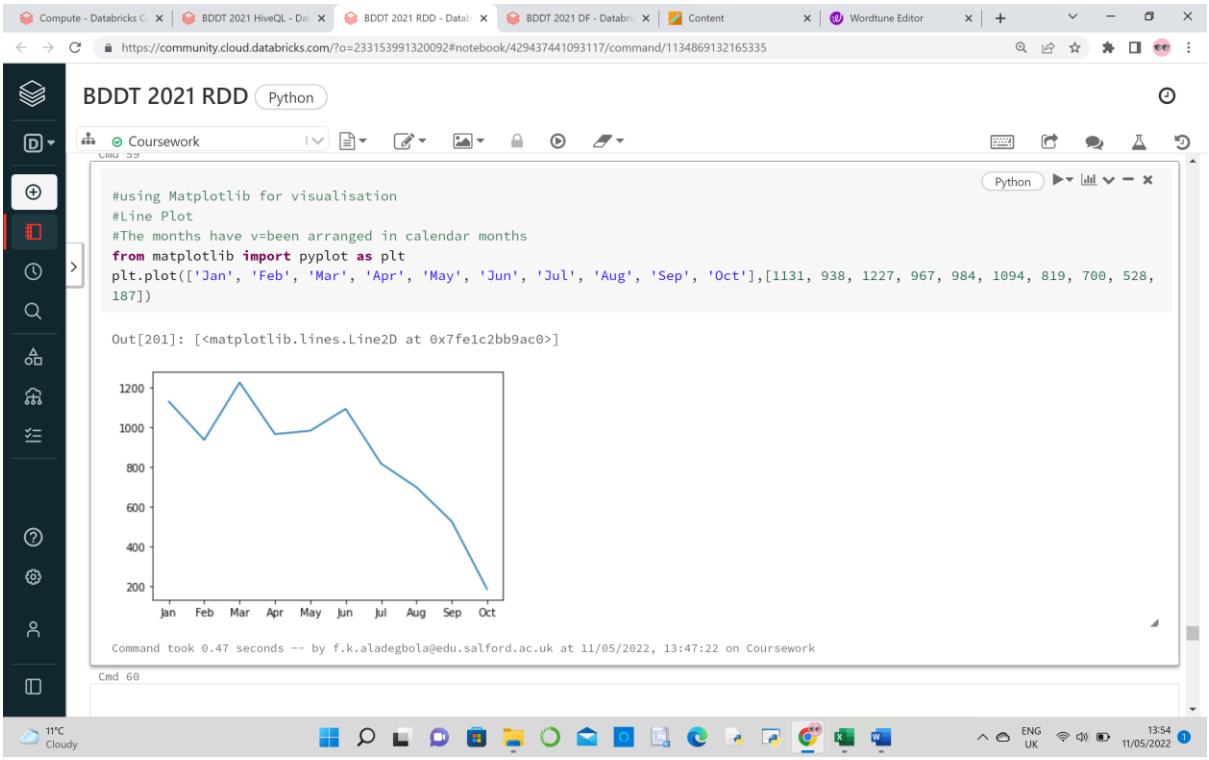
Matplotlib is a Python package that facilitates creating static, animated, and interactive #visualizationsThe x-axis and y-axis were selected as shown below.



The bar plot in rdd: Matplotlib was used for visualisation (Bar Plot)

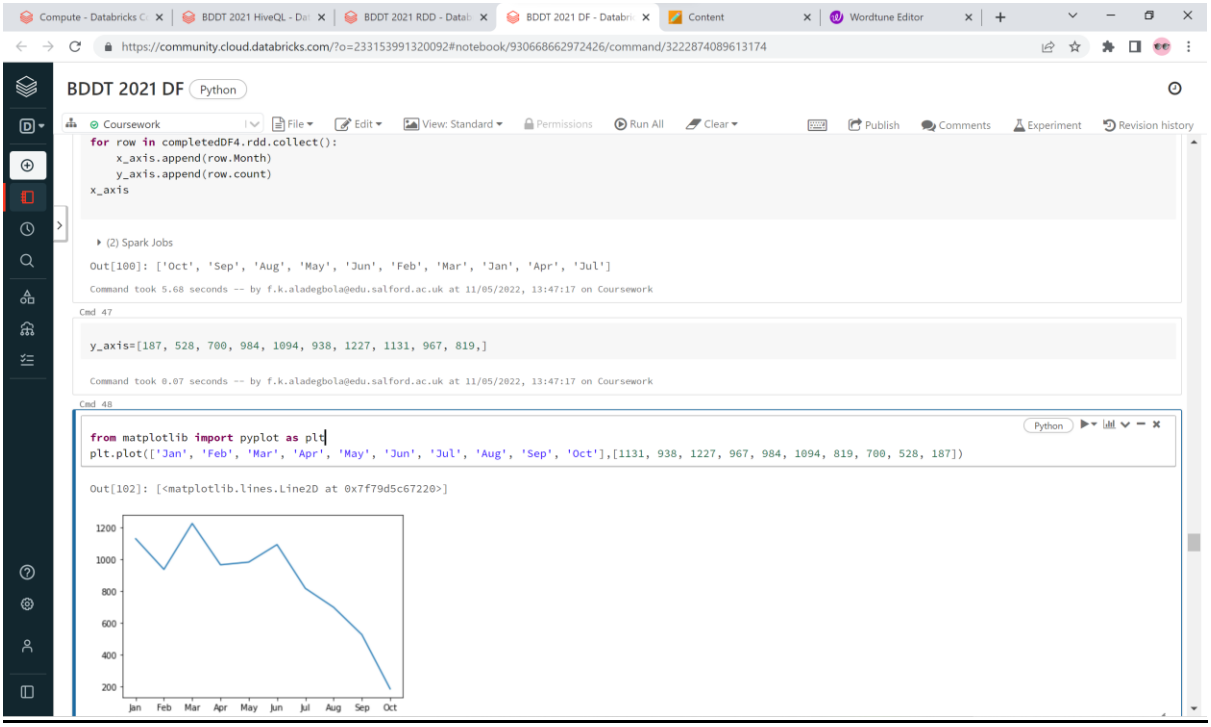


The Line Plot in rdd: Matplotlib

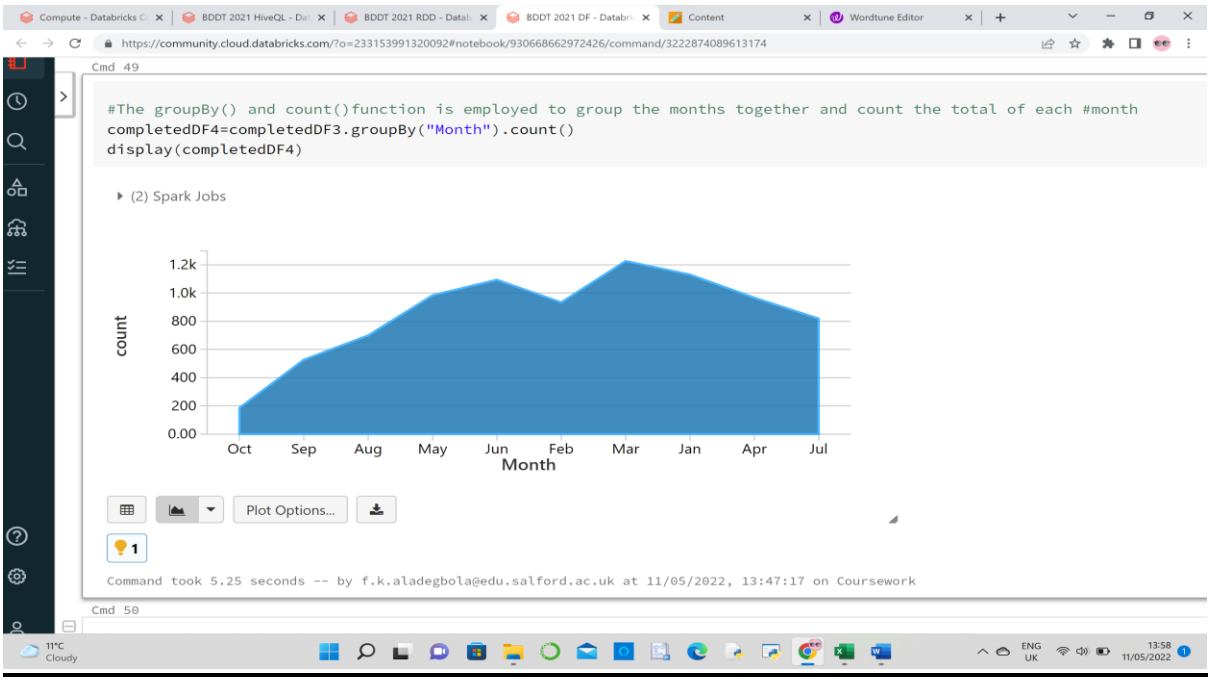


DATAFRAME: Matplotlib was used for visualisation

The Line plot in dataframe showed March as the highest month of completed clinical studies in 2021. It has the highest point on the line plot.



The Area Inbuilt Visualization in Dataframe

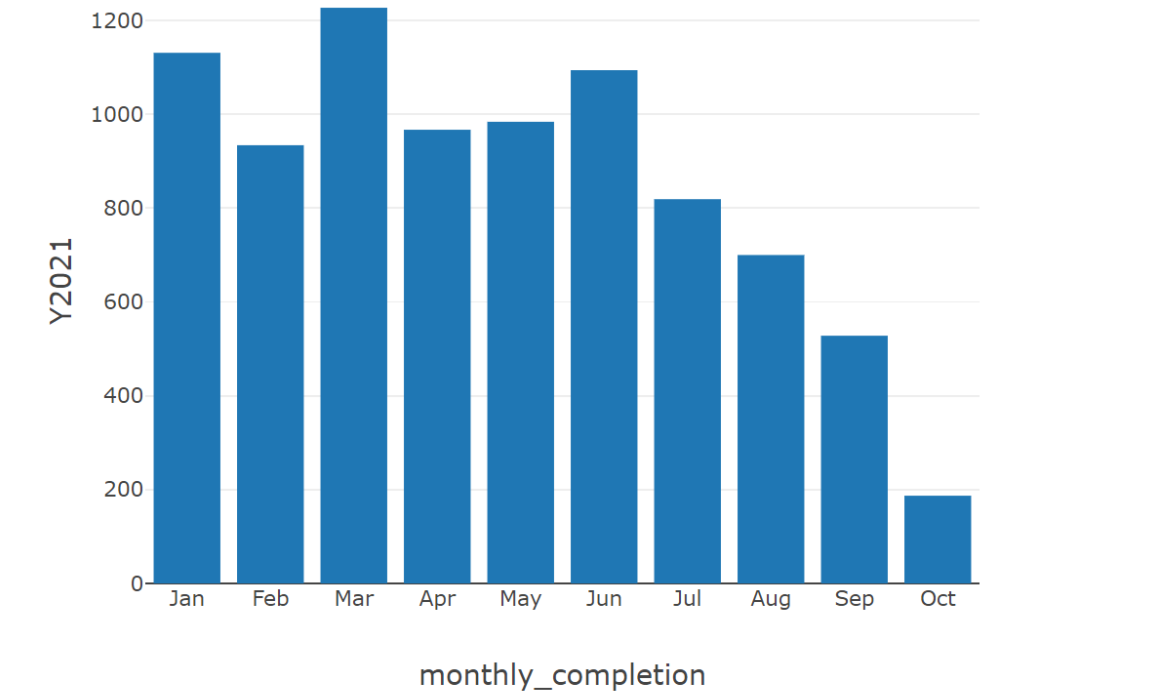


HIVEQL

The Area and Pie chart Inbuilt visualization was used in Hiveql

The bar chart showed March as the month with the highest number of completed clinical studies in 2021.

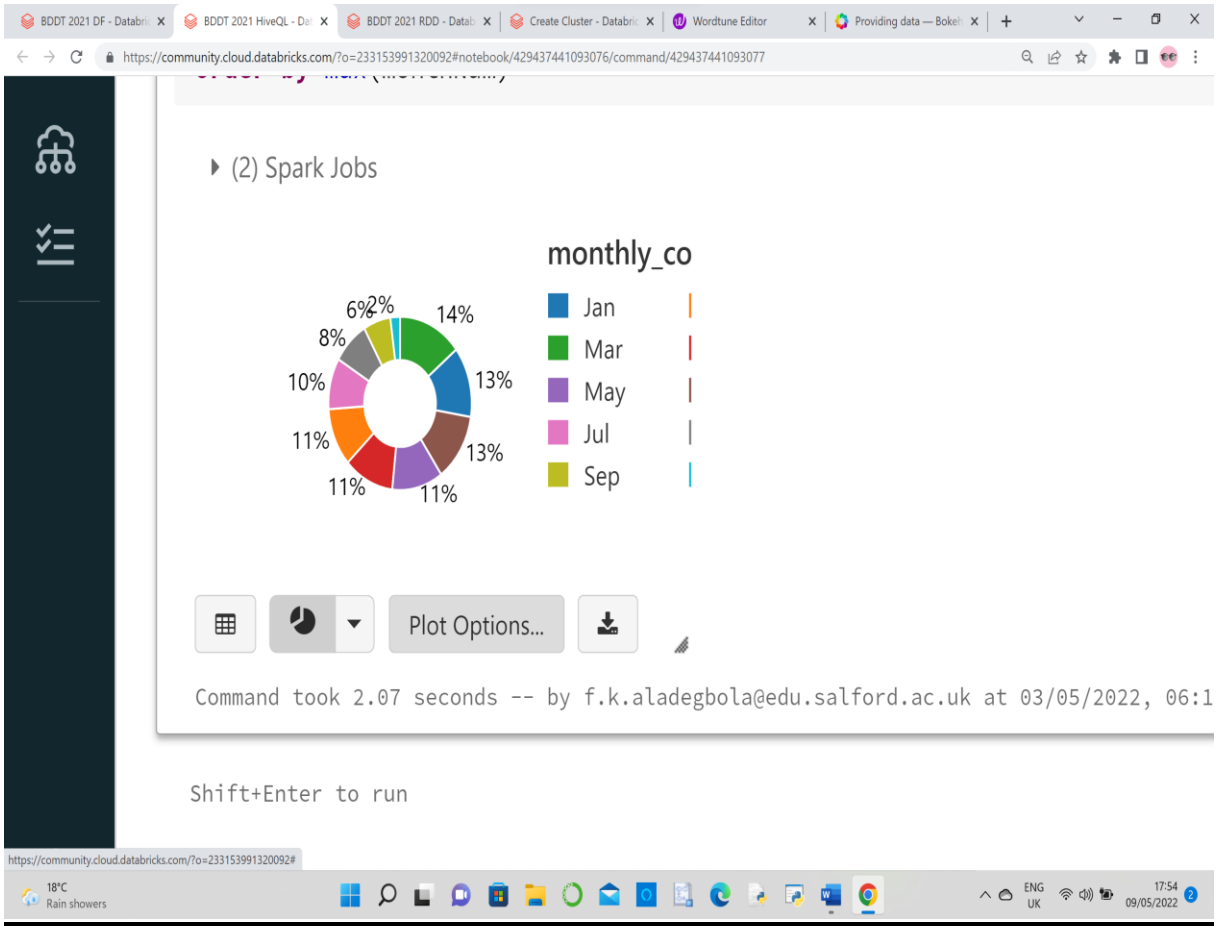
(2) Spark Jobs



Plot Options...

Command took 1.60 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 06:10:50 on COURSEWORK

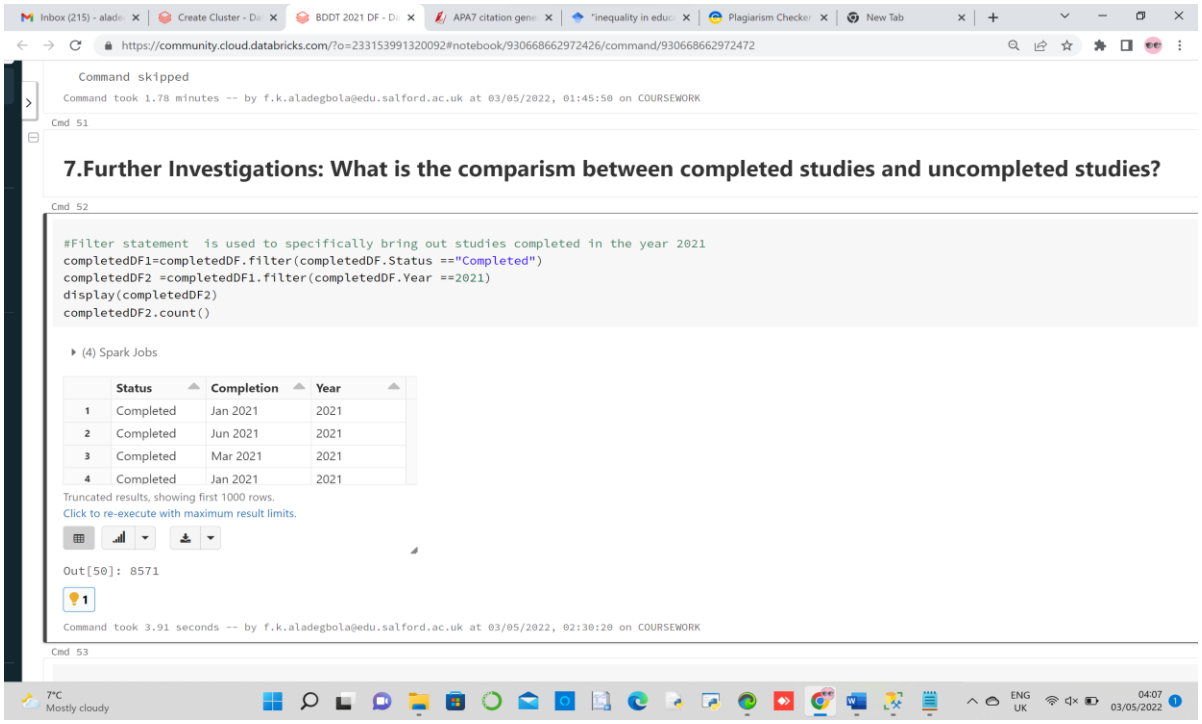
The pie chart showed March as the highest month of a completed clinical study by assigning the highest percentile 14%.

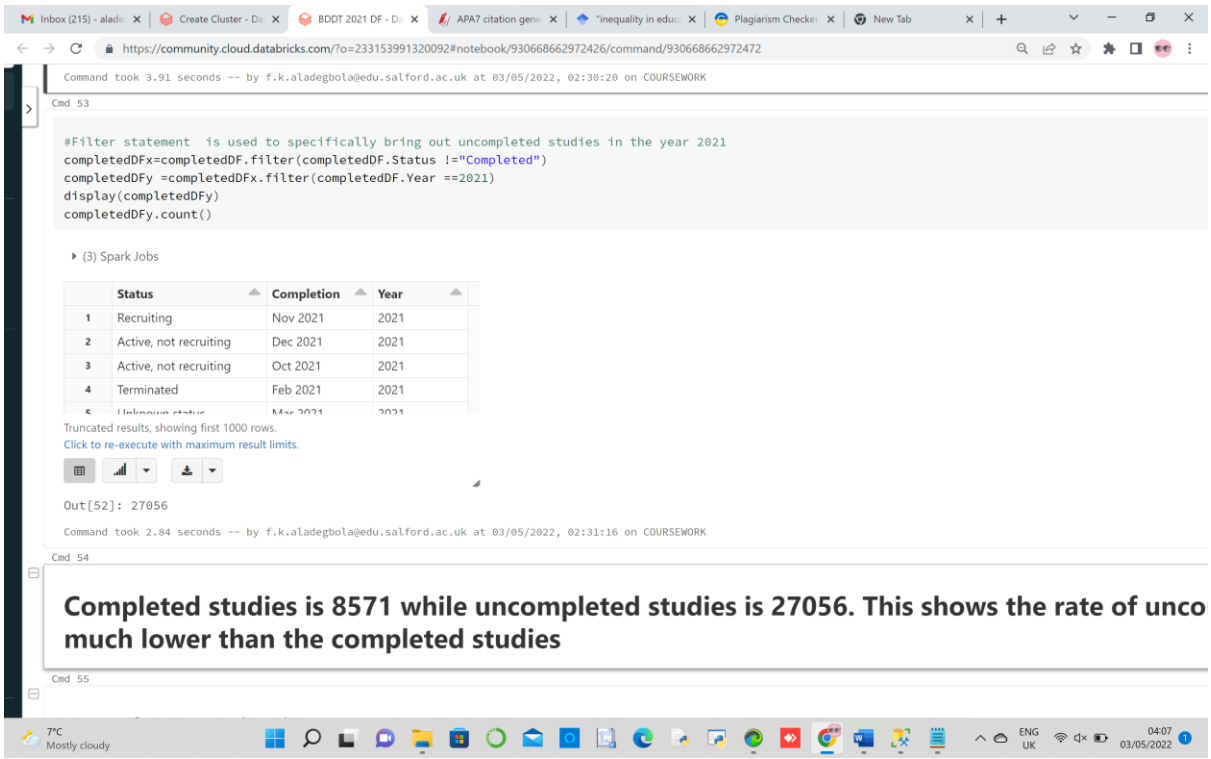


FURTHER ANALYSIS 1

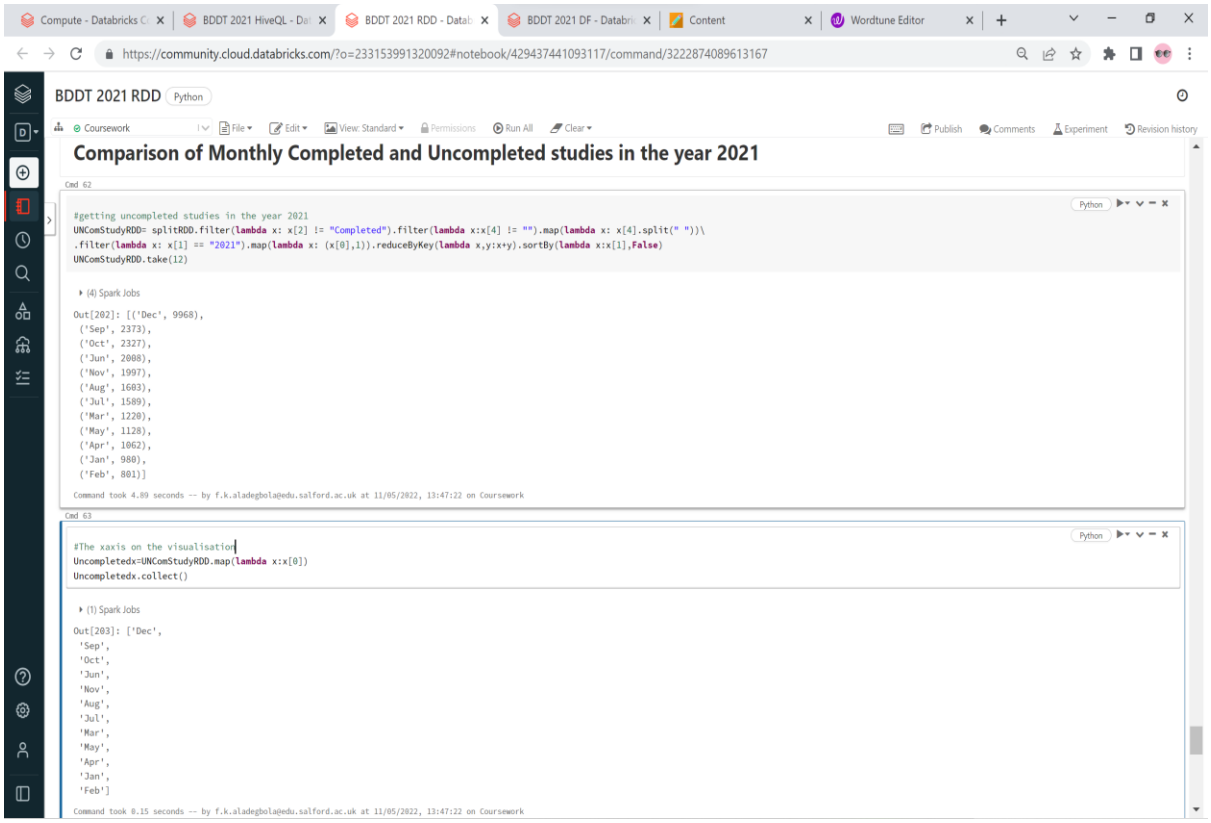
DATAFRAME

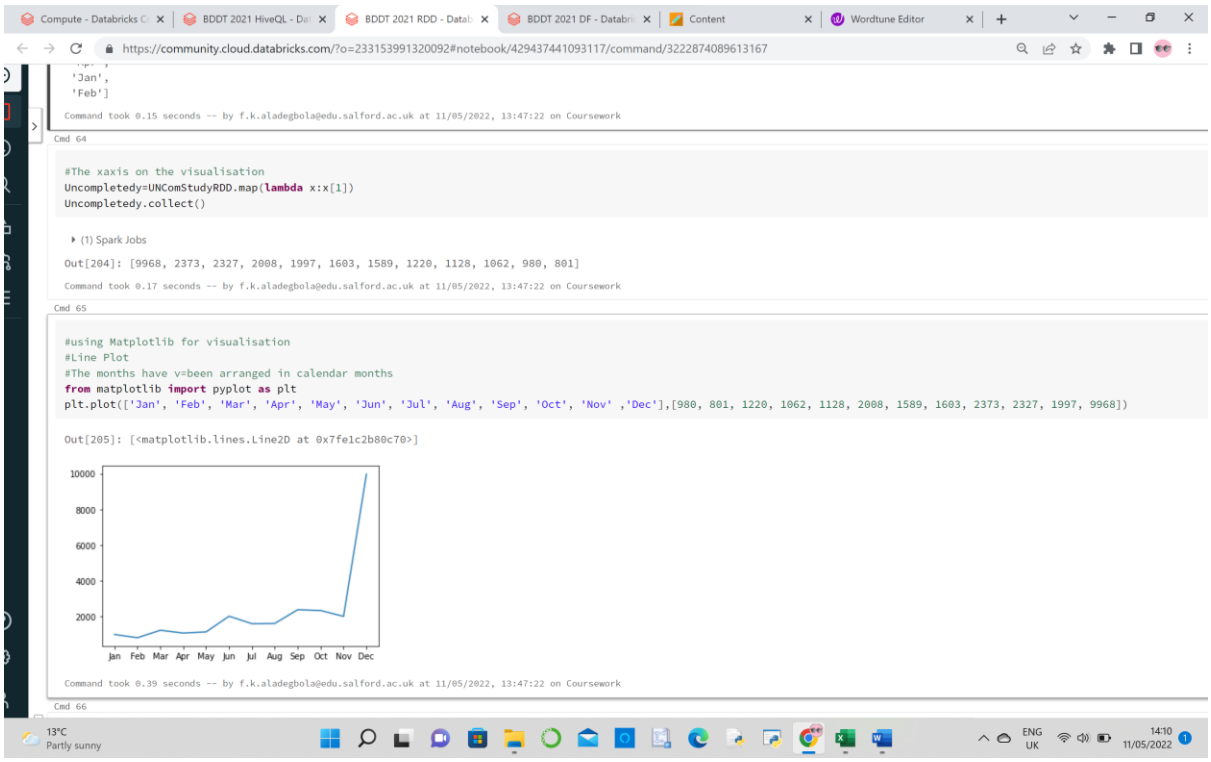
Comparison between completed studies and uncompleted studies in terms of count. A count of completed studies in 2021 was done against the uncompleted studies. Completed studies showed a count of 8571 while the uncompleted study showed a count of 27,056. This shows that more effort should be geared towards more studies completed. There is a need for research into the reason for a lower number of the completed study.





RDD IMPLEMETATION


BDDT 2021 RDD
Python
Comparison of Monthly Completed and Uncompleted studies in the year 2021
Cmd 62
#getting uncompleted studies in the year 2021
UNComStudyRDD= splitRDD.filter(lambda x: x[2] != "Completed").filter(lambda x:x[4] != "").map(lambda x: x[4].split(" "))\
.filter(lambda x: x[1] == "2021").map(lambda x: (x[0],1)).reduceByKey(lambda x,y:x+y).sortBy(lambda x:x[1],False)
UNComStudyRDD.take(12)
(4) Spark Jobs
Out[202]: [('Dec', 9968),
('Sep', 2373),
('Oct', 2327),
('Jun', 2008),
('Nov', 1997),
('Aug', 1603),
('Jul', 1589),
('Mar', 1220),
('May', 1128),
('Apr', 1062),
('Jan', 980),
('Feb', 801)]
Command took 4.89 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 11/05/2022, 13:47:22 on Coursework
Cmd 63
#The xaxis on the visualisation
Uncompletedx=UNComStudyRDD.map(lambda x:x[0])
Uncompletedx.collect()
(1) Spark Jobs
Out[203]: ['Dec',
'Sep',
'Oct',
'Jun',
'Nov',
'Aug',
'Jul',
'Mar',
'May',
'Apr',
'Jan',
'Feb']
Command took 0.15 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 11/05/2022, 13:47:22 on Coursework

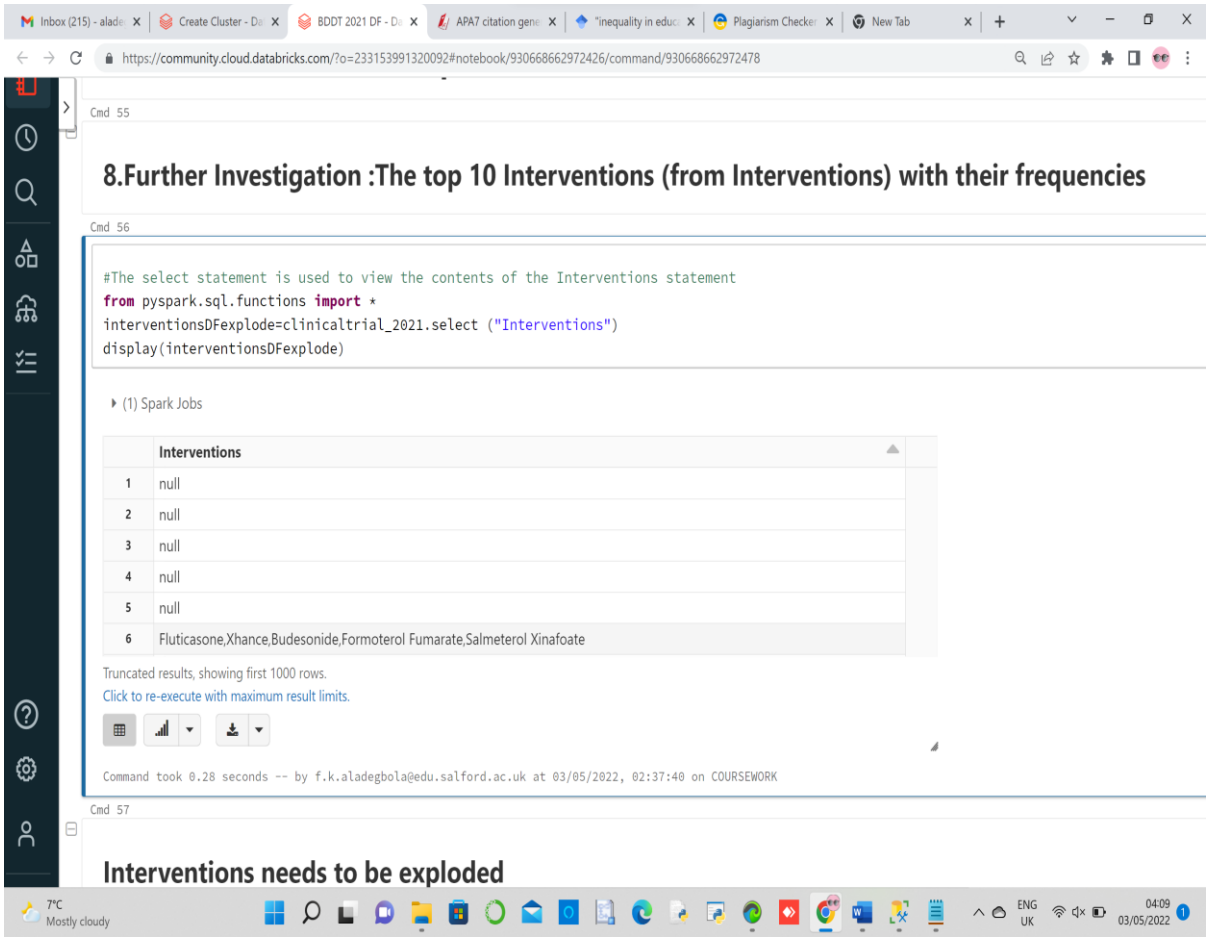

'Jan',
'Feb']
Command took 0.15 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 11/05/2022, 13:47:22 on Coursework
Cmd 64
#The xaxis on the visualisation
Uncompletedy=UNComStudyRDD.map(lambda x:x[1])
Uncompletedy.collect()
(1) Spark Jobs
Out[204]: [9968, 2373, 2327, 2008, 1997, 1603, 1589, 1220, 1128, 1062, 980, 801]
Command took 0.17 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 11/05/2022, 13:47:22 on Coursework
Cmd 65
#using Matplotlib for visualisation
#Line Plot
#The months have v=been arranged in calendar months
from matplotlib import pyplot as plt
plt.plot(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov' ,'Dec'],[980, 801, 1220, 1062, 1128, 2008, 1589, 1603, 2373, 2327, 1997, 9968])
Out[205]: [<matplotlib.lines.Line2D at 0x7fe1c2b80c70>]
10000
8000
6000
4000
2000
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Command took 0.39 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 11/05/2022, 13:47:22 on Coursework
Cmd 66

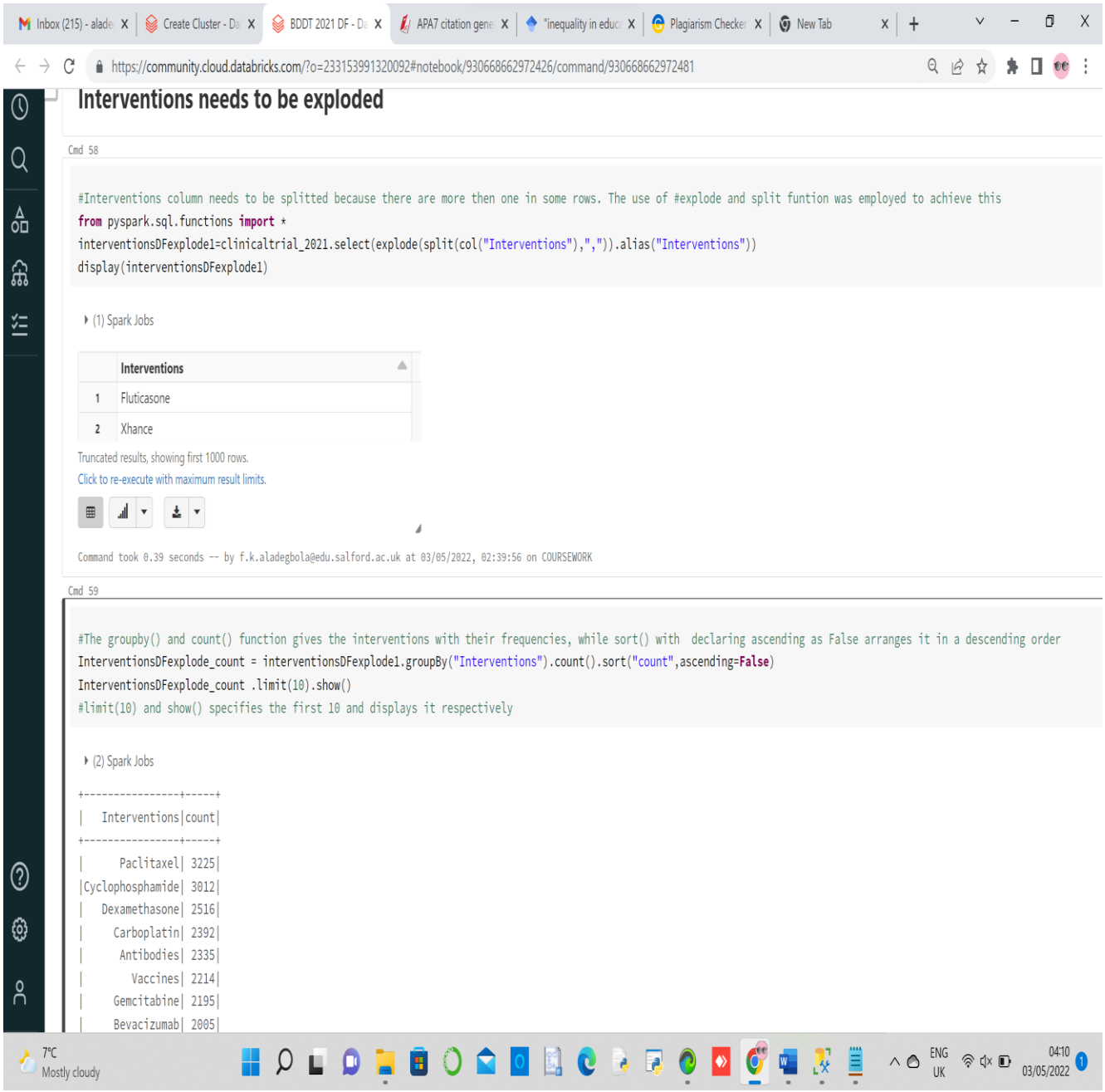
There were no completed studies in November and December 2021, hence as shown in the line plot above uncompleted studies reached its peak in November and December 2021. It is advised that a feasibility study should be put in place to check the cause of a high rate of uncompletion at the end of the year and measures should be put in place to mitigate this.

FURTHER ANALYSIS 2

DATAFRAME

What are the top 10 Interventions from Interventions and their frequencies?



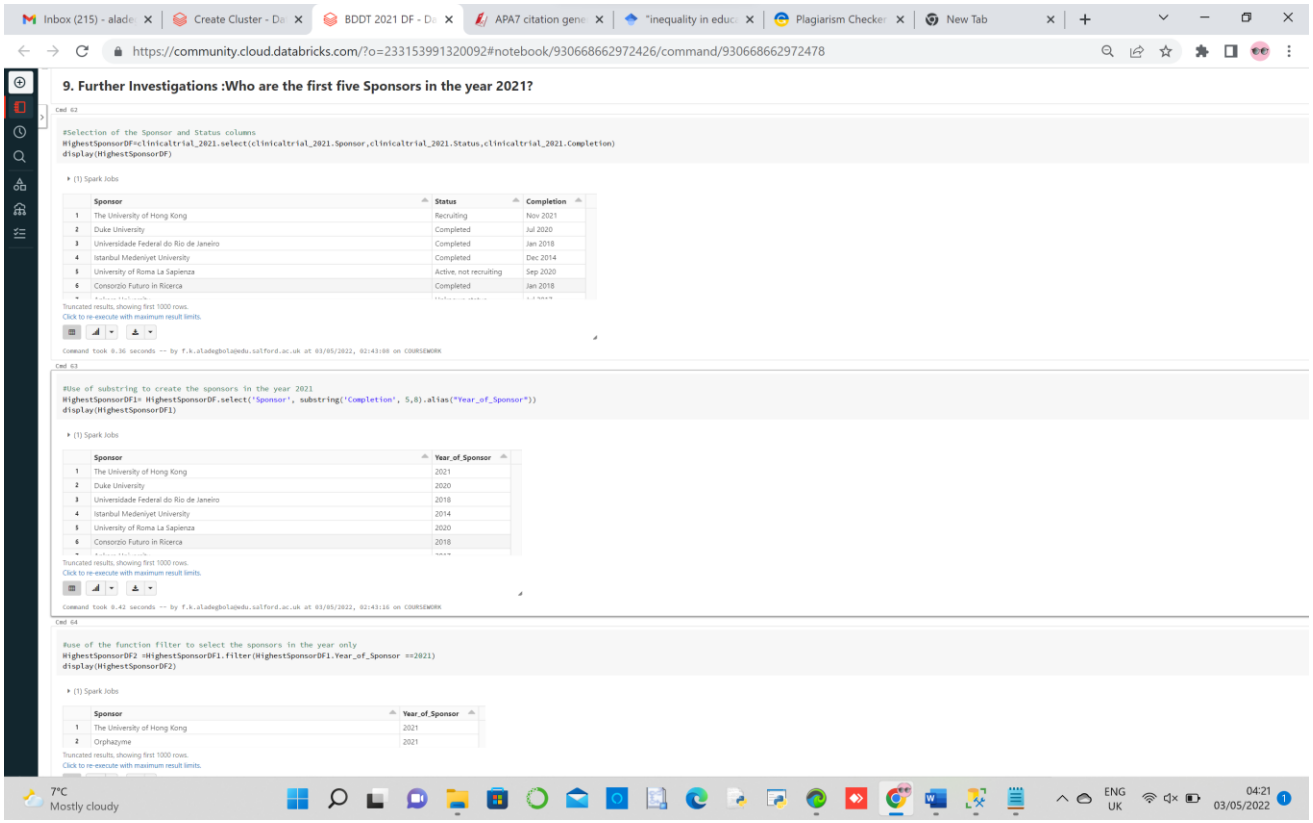


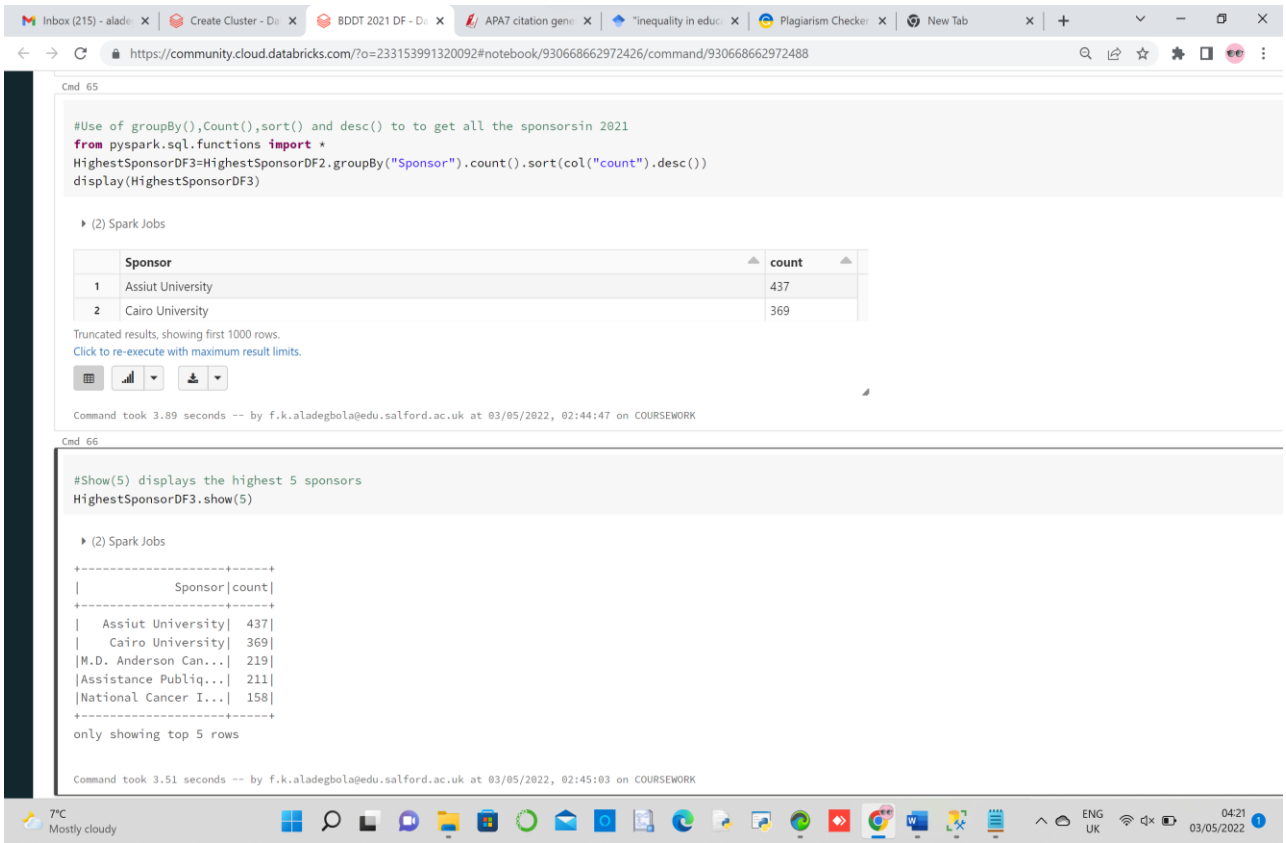
FURTHER ANALYSIS 3

DATAFRAME

Who are the first 5 sponsors in the year 2021 with the highest frequency?

Assiut University emerged as the highest sponsor in 2021 with a count of 447 clinical trials sponsored.


9. Further Investigations :Who are the first five Sponsors in the year 2021?
#Selection of the Sponsor and Status columns
HighestSponsorDF=clinicaltrial_2021.select(clinicaltrial_2021.Sponsor,clinicaltrial_2021.Status,clinicaltrial_2021.Completion)
display(HighestSponsorDF)
(1) Spark Jobs
Sponsor Status Completion
1 The University of Hong Kong Recruiting Nov 2021
2 Duke University Completed Jul 2020
3 Universidade Federal do Rio de Janeiro Completed Jan 2018
4 Istanbul Medeniyet University Completed Dec 2014
5 University of Roma La Sapienza Active, not recruiting Sep 2020
6 Consorzio Futuro in Ricerca Completed Jan 2018
Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.
Command took 0.36 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 02:43:08 on COURSEWORK
#Use of substring to create the sponsors in the year 2021
HighestSponsorDF1= HighestSponsorDF.select('Sponsor', substring('Completion', 5,8).alias("Year_of_Sponsor"))
display(HighestSponsorDF1)
(1) Spark Jobs
Sponsor Year_of_Sponsor
1 The University of Hong Kong 2021
2 Duke University 2020
3 Universidade Federal do Rio de Janeiro 2018
4 Istanbul Medeniyet University 2014
5 University of Roma La Sapienza 2020
6 Consorzio Futuro in Ricerca 2018
Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.
Command took 0.42 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 02:43:16 on COURSEWORK
#use of the function filter to select the sponsors in the year only
HighestSponsorDF2 =HighestSponsorDF1.filter(HighestSponsorDF1.Year_of_Sponsor ==2021)
display(HighestSponsorDF2)
(1) Spark Jobs
Sponsor Year_of_Sponsor
1 The University of Hong Kong 2021
2 Orphazyme 2021
Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.


#Use of groupBy(),Count(),sort() and desc() to to get all the sponsorsin 2021
from pyspark.sql.functions import *
HighestSponsorDF3=HighestSponsorDF2.groupBy("Sponsor").count().sort(col("count").desc())
display(HighestSponsorDF3)
(2) Spark Jobs
Sponsor count
1 Assiut University 437
2 Cairo University 369
Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.
Command took 3.89 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 02:44:47 on COURSEWORK
#Show(5) displays the highest 5 sponsors
HighestSponsorDF3.show(5)
(2) Spark Jobs
+--------------------+-----+
| Sponsor|count|
+--------------------+-----+
| Assiut University| 437|
| Cairo University| 369|
|M.D. Anderson Can...| 219|
|Assistance Publiq...| 211|
|National Cancer I...| 158|
+--------------------+-----+
only showing top 5 rows
Command took 3.51 seconds -- by f.k.aladegbola@edu.salford.ac.uk at 03/05/2022, 02:45:03 on COURSEWORK