CLOUD TECHNOLOGIES AND BIG DATA FRAMEWORKS

3rd Assignment – Covid-19 Cases in Italy 2020

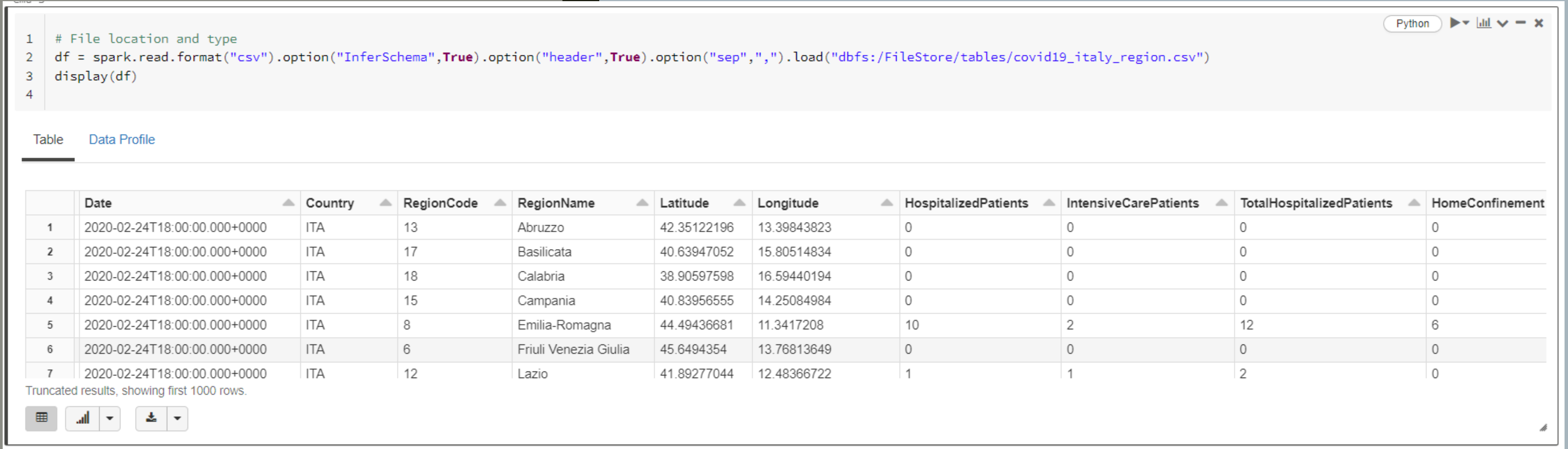
Overview

The dataset that has been retrieved contains a csv file that includes information regarding Covid-19 cases tracking back from 24-02-2020 until 11-04-2020.

In this notebook I will conduct EDA on uploaded table using Data-Frames APIs and SQL query language, combined with Data Visualization to get a better insight on the data available to test. There won’t be data integration and further processing (train/test/split and cross validations) to implement ML algorithms to try test the accuracy of different predictive time series models, since the dataset is 2 years old.

Uploading the Data

Initially two commands are executed. We want to see the file path in order to track the file that we want to upload and then we create a data frame with the name df.



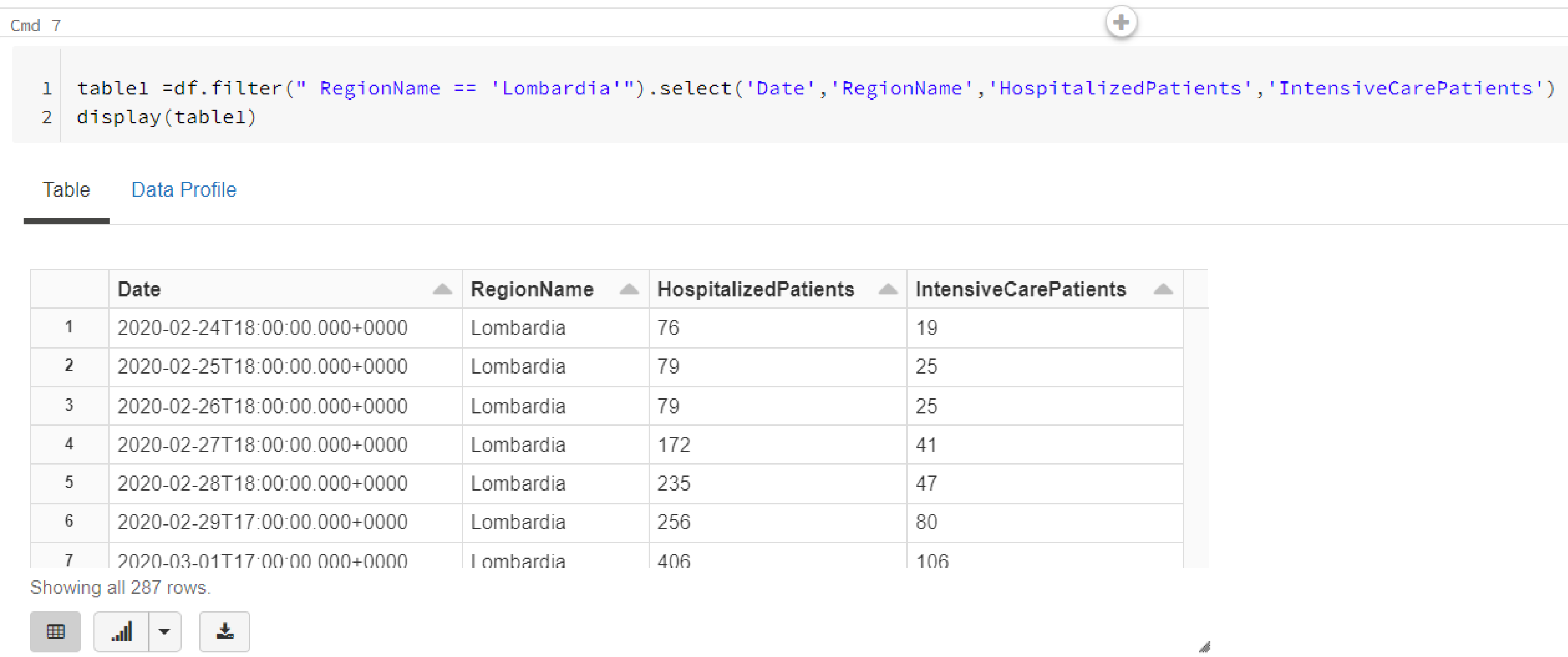
```
# File location and type
df = spark.read.format("csv").option("InferSchema",True).option("header",True).option("sep",",").load("dbfs:/FileStore/tables/covid19_italy_region.csv")
display(df)

```

	Date	Country	RegionCode	RegionName	Latitude	Longitude	HospitalizedPatients	IntensiveCarePatients	TotalHospitalizedPatients	HomeConfinement
1	2020-02-24T18:00:00.000+0000	ITA	13	Abruzzo	42.35122196	13.39843823	0	0	0	0
2	2020-02-24T18:00:00.000+0000	ITA	17	Basilicata	40.63947052	15.80514834	0	0	0	0
3	2020-02-24T18:00:00.000+0000	ITA	18	Calabria	38.90597598	16.59440194	0	0	0	0
4	2020-02-24T18:00:00.000+0000	ITA	15	Campania	40.83956555	14.25084984	0	0	0	0
5	2020-02-24T18:00:00.000+0000	ITA	8	Emilia-Romagna	44.49436681	11.3417208	10	2	12	6
6	2020-02-24T18:00:00.000+0000	ITA	6	Friuli Venezia Giulia	45.6494354	13.76813649	0	0	0	0
7	2020-02-24T18:00:00.000+0000	ITA	12	Lazio	41.89277044	12.48366722	1	1	2	0

Truncated results, showing first 1000 rows.

Test 1

First I want to filter my dataset to test and see for each day that data was collected how many of the patients that were hospitalized were categorized as intensive care patients for the region of Lombardia in Italy. Also get insight on how these number changed over time and create my second dataframe.



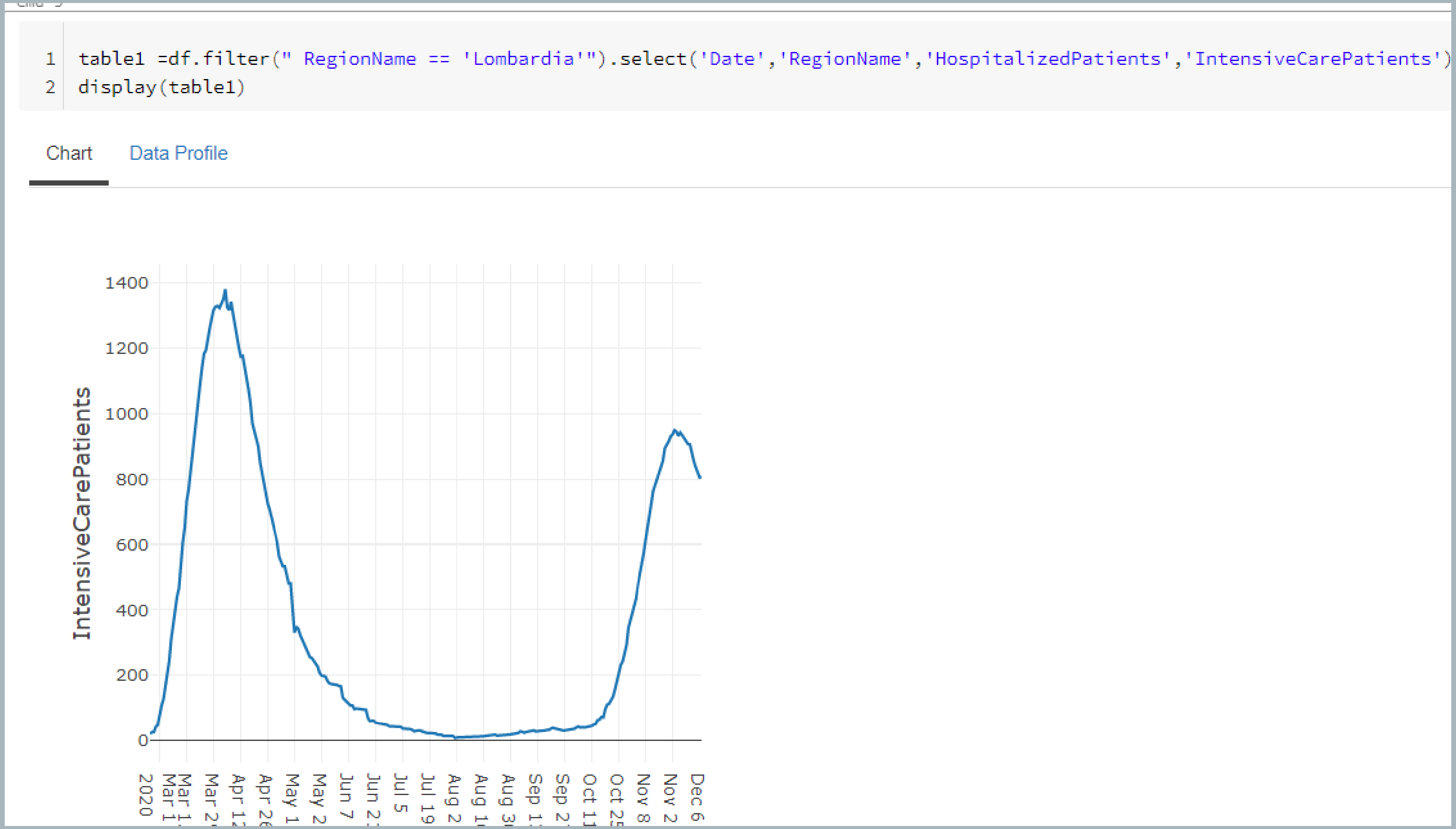
```
table1 =df.filter(" RegionName == 'Lombardia'").select('Date','RegionName','HospitalizedPatients','IntensiveCarePatients')
display(table1)
```

	Date	RegionName	HospitalizedPatients	IntensiveCarePatients
1	2020-02-24T18:00:00.000+0000	Lombardia	76	19
2	2020-02-25T18:00:00.000+0000	Lombardia	79	25
3	2020-02-26T18:00:00.000+0000	Lombardia	79	25
4	2020-02-27T18:00:00.000+0000	Lombardia	172	41
5	2020-02-28T18:00:00.000+0000	Lombardia	235	47
6	2020-02-29T17:00:00.000+0000	Lombardia	256	80
7	2020-03-01T17:00:00.000+0000	Lombardia	406	106

Showing all 287 rows.

Graph 1

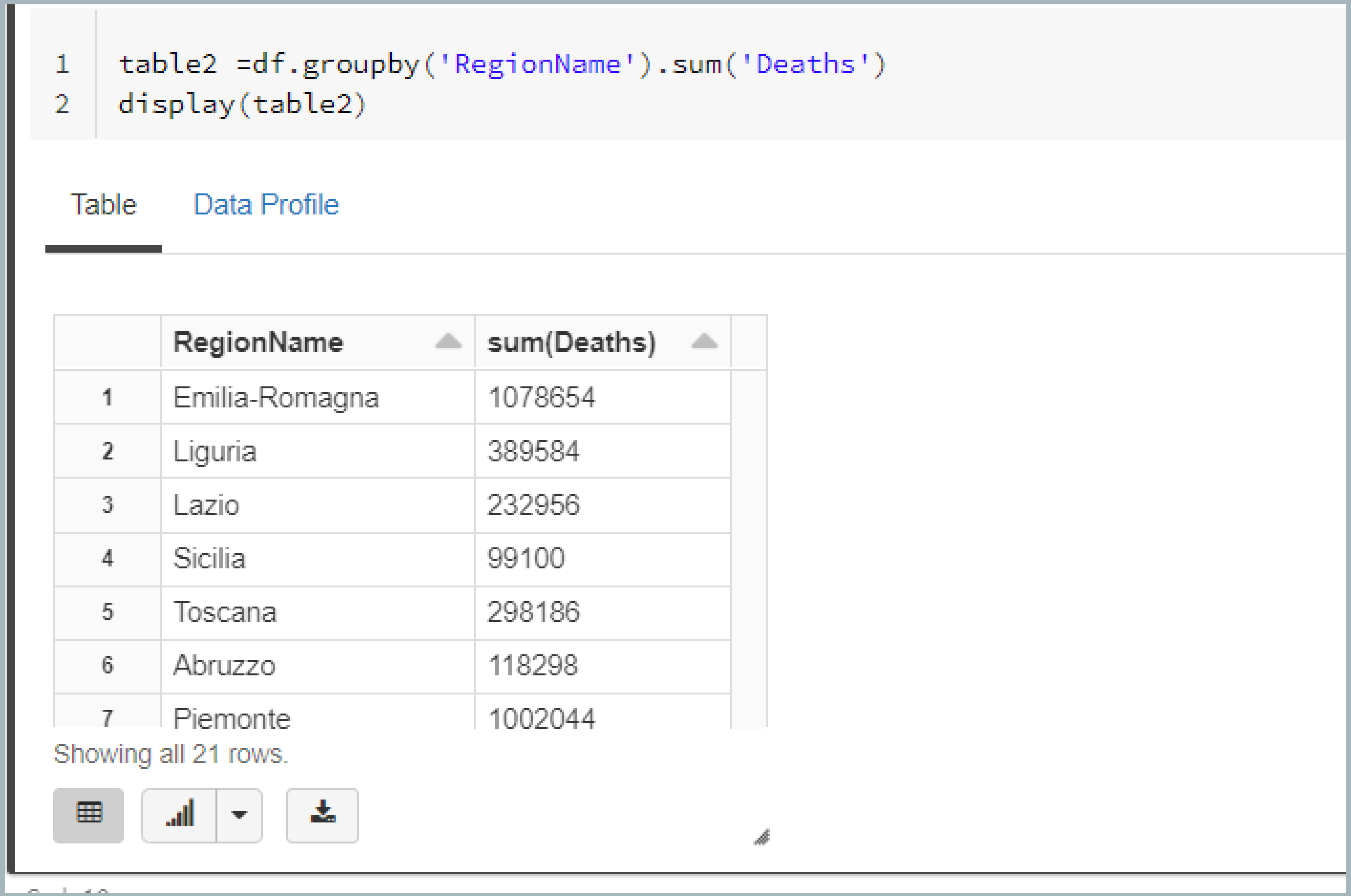
Graphical representation using a line graph (that is best suited for visualizing continuous time variables) to see how Intensive Care Patient's curve fluctuates over this time span



```
table1 =df.filter(" RegionName == 'Lombardia'").select('Date','RegionName','HospitalizedPatients','IntensiveCarePatients')
display(table1)
```

Test 2

Let’s say now that I want to compare the number of deaths recorded in all regions



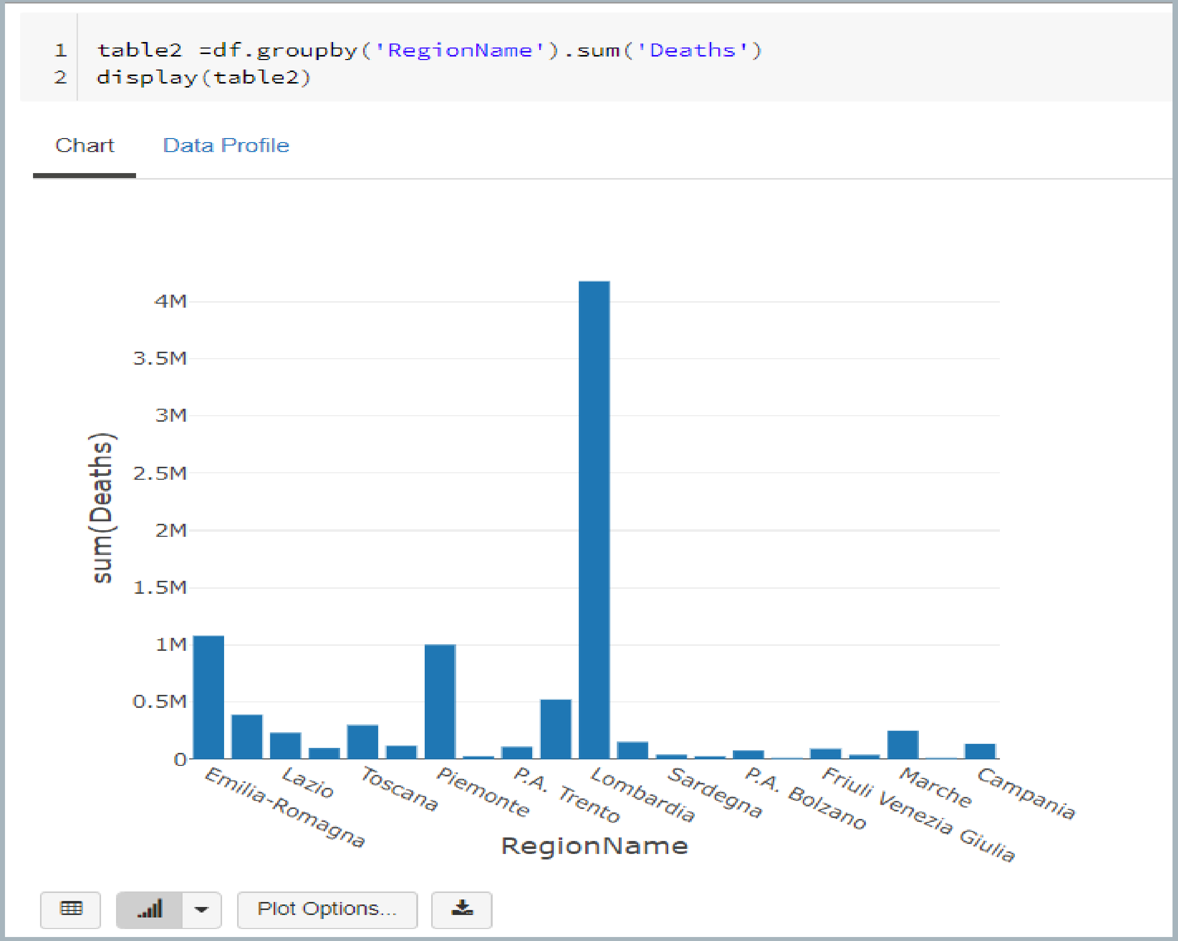
```
table2 =df.groupby('RegionName').sum('Deaths')
display(table2)
```

	RegionName	sum(Deaths)
1	Emilia-Romagna	1078654
2	Liguria	389584
3	Lazio	232956
4	Sicilia	99100
5	Toscana	298186
6	Abruzzo	118298
7	Piemonte	1002044

Showing all 21 rows.

Graph 2

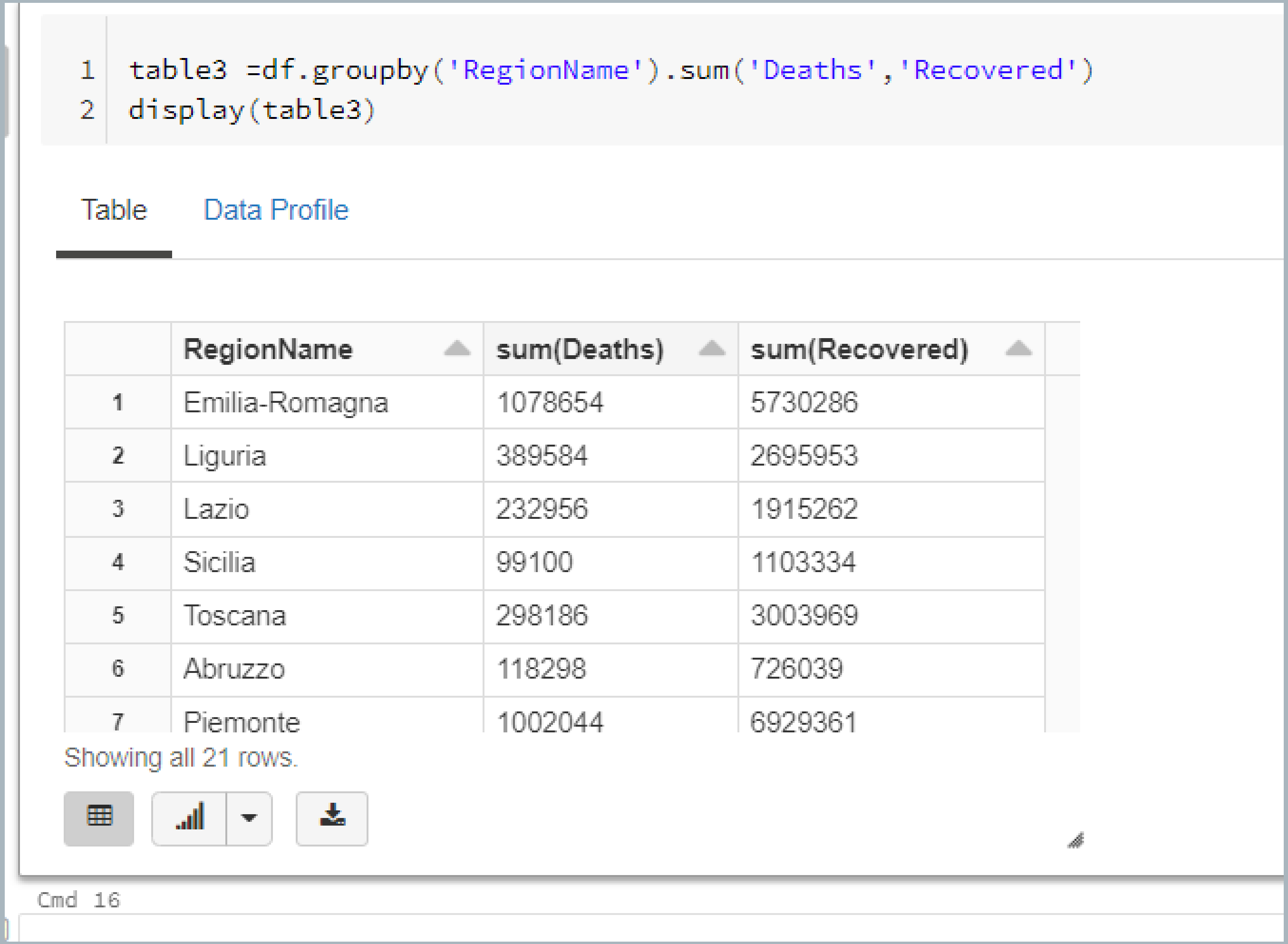
Use a bar chart to create a clear visual of the table2 dataframe created above to showcase the comparison between all Italy regions regarding the total number of deaths



```
table2 =df.groupby('RegionName').sum('Deaths')
display(table2)
```

Test 3

Compare how many of the patients that were hospitalized managed to recover and what amount of these patients actually died, for all regions in Italy.



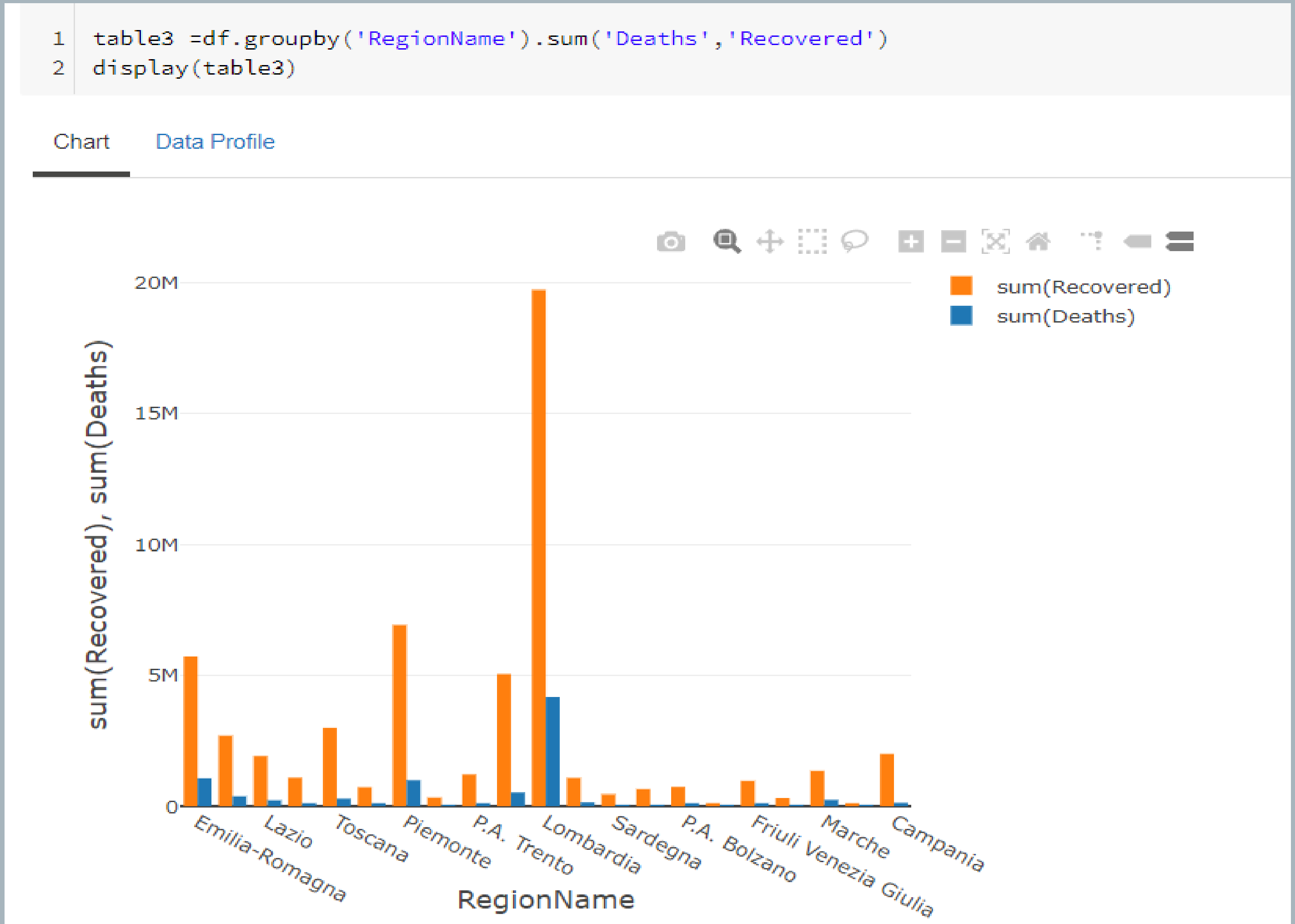
```
table3 =df.groupby('RegionName').sum('Deaths','Recovered')
display(table3)
```

	RegionName	sum(Deaths)	sum(Recovered)
1	Emilia-Romagna	1078654	5730286
2	Liguria	389584	2695953
3	Lazio	232956	1915262
4	Sicilia	99100	1103334
5	Toscana	298186	3003969
6	Abruzzo	118298	726039
7	Piemonte	1002044	6929361

Showing all 21 rows.

Graph 3

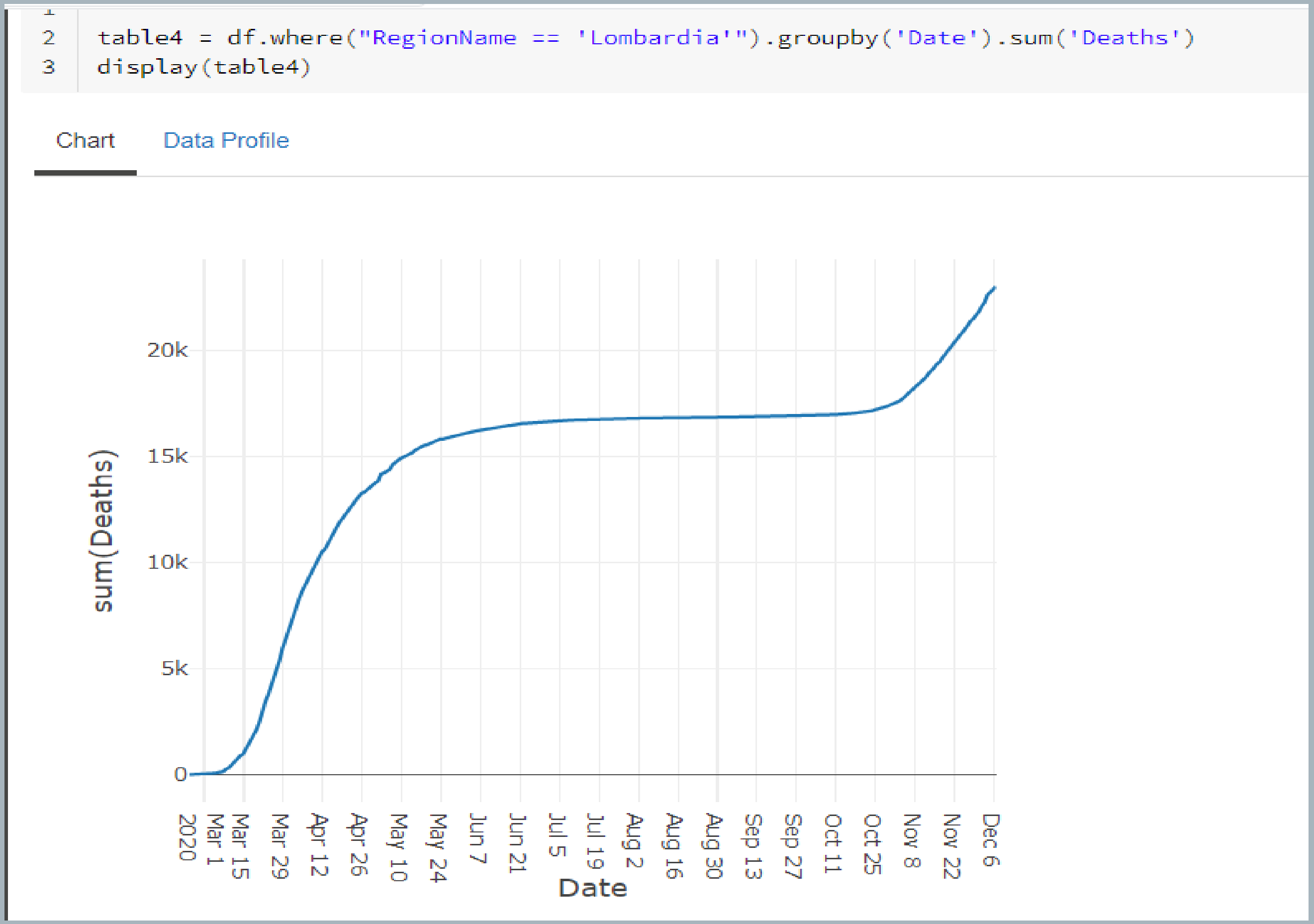
Graphical representation of the comparison between the total number of people that died and people that recovered with respect to the different regions.



```
table3 =df.groupby('RegionName').sum('Deaths','Recovered')
display(table3)
```

Test 4

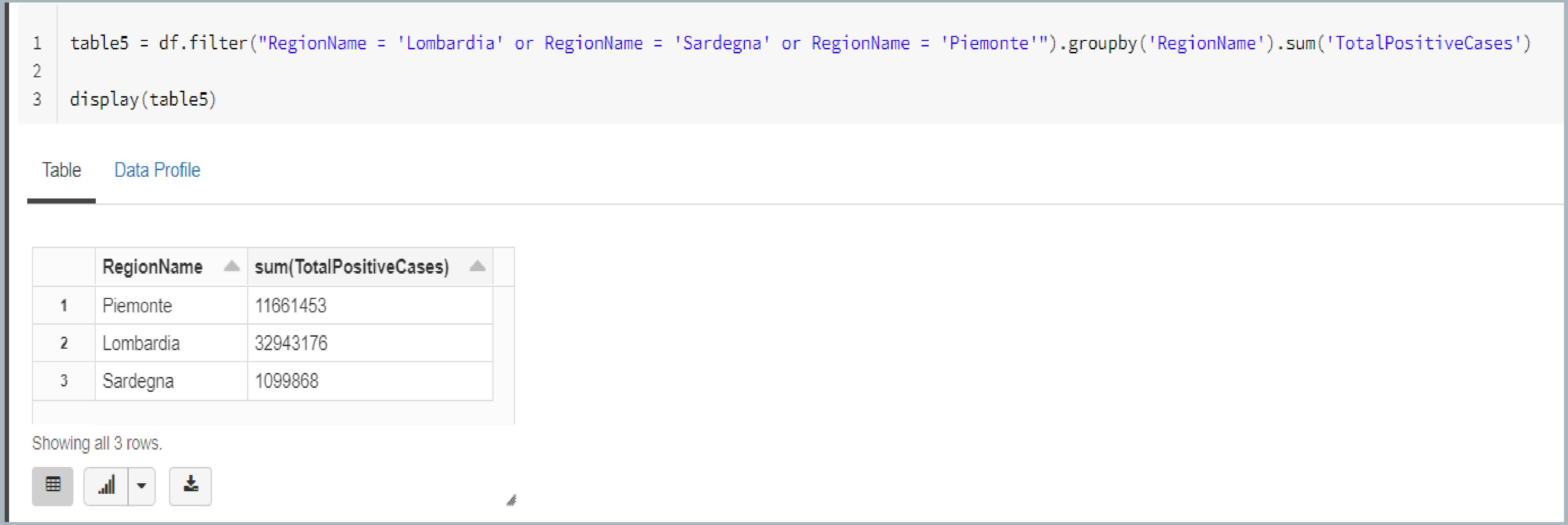
Since we can see from the graph above that in Lombardia we have the largest number of deaths for patients that were hospitalized, we can further create another visual to see which time period we had the highest number of deaths occurred.



```
table4 = df.where("RegionName == 'Lombardia'").groupby('Date').sum('Deaths')
display(table4)
```

Test 5

Now we can compare the sum of total positive cases in the regions of Lombardia, Sardegna and Piemonte

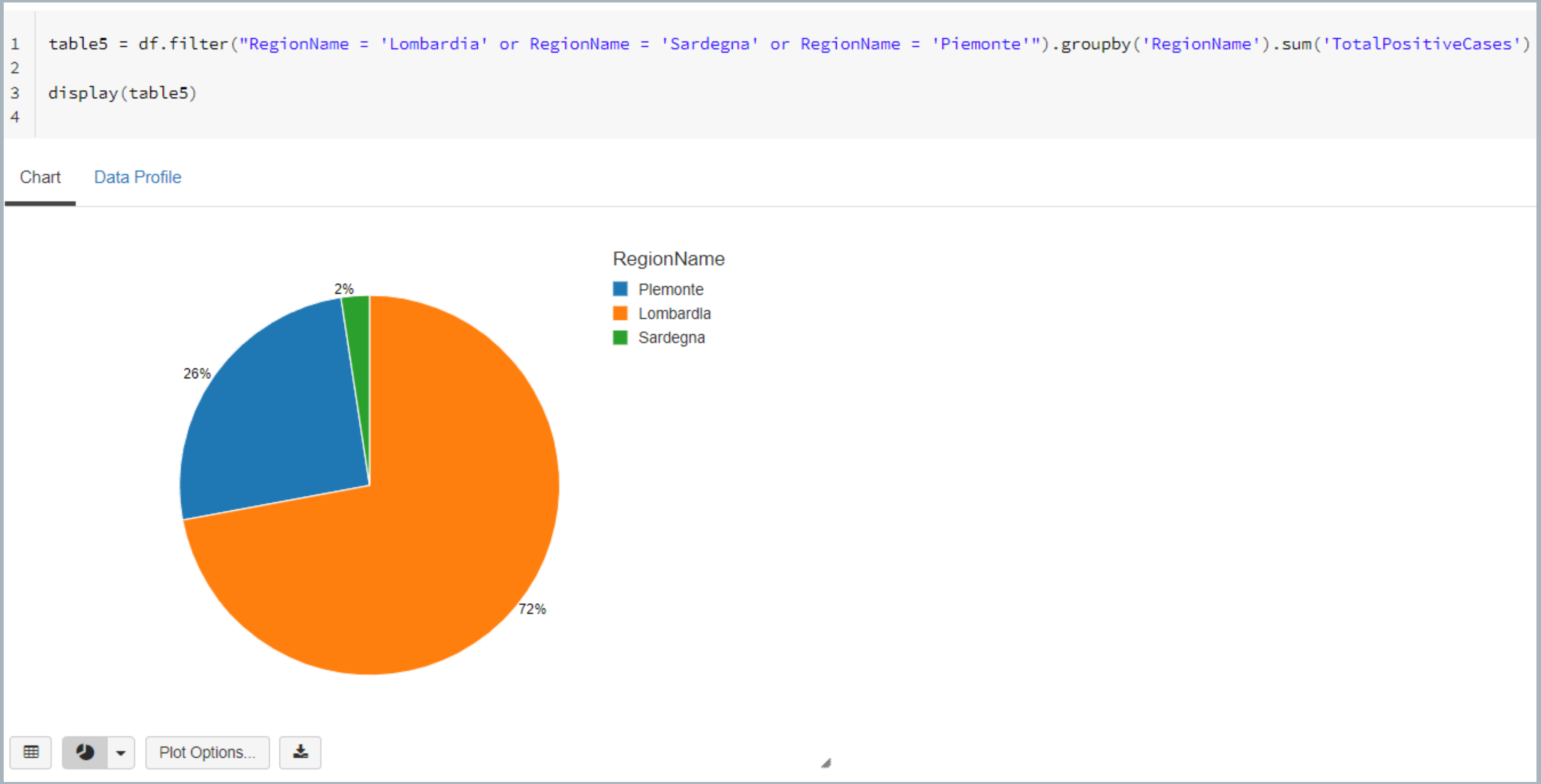


```
table5 = df.filter("RegionName = 'Lombardia' or RegionName = 'Sardegna' or RegionName = 'Piemonte'").groupby('RegionName').sum('TotalPositiveCases')

display(table5)
```

	RegionName	sum(TotalPositiveCases)
1	Piemonte	11661453
2	Lombardia	32943176
3	Sardegna	1099868

Showing all 3 rows.

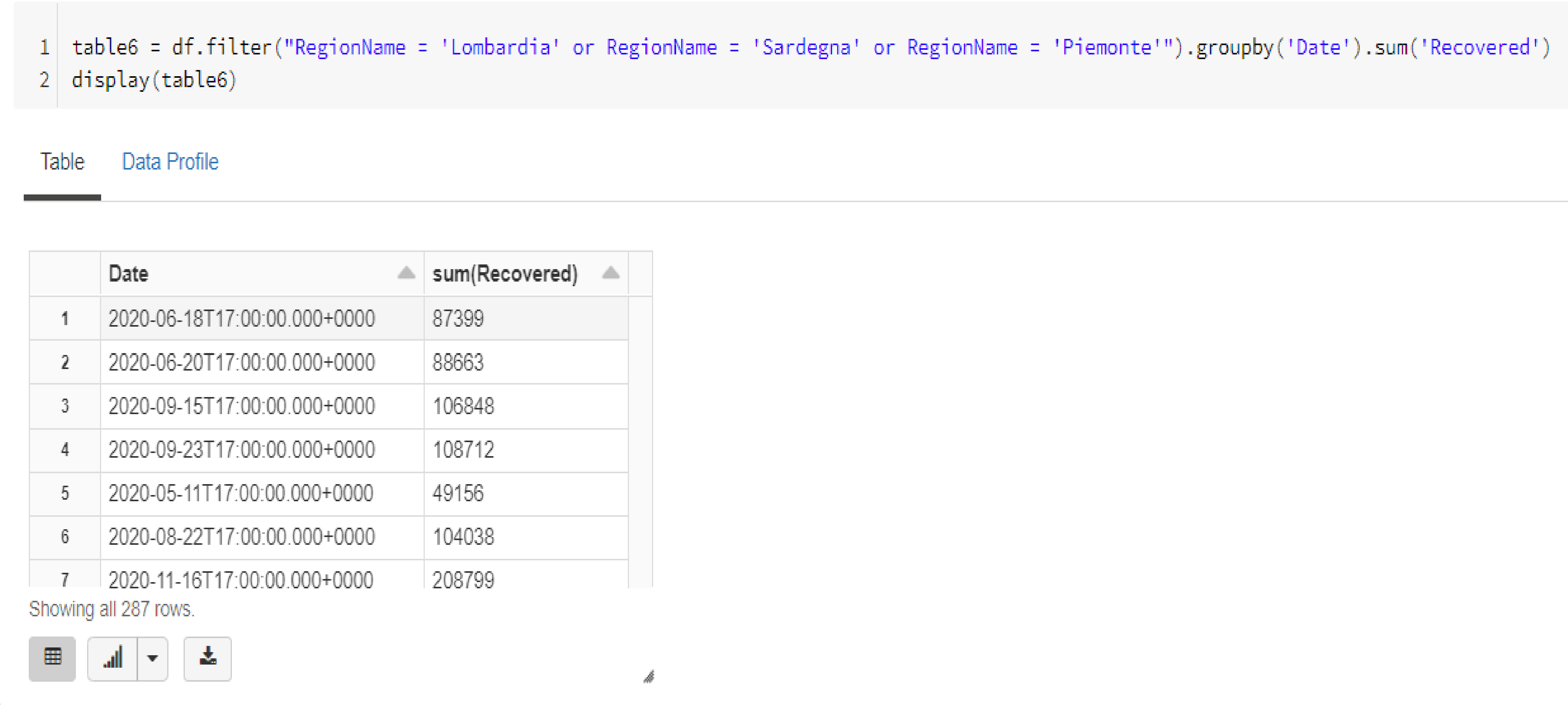
Graph 5



```
table5 = df.filter("RegionName = 'Lombardia' or RegionName = 'Sardegna' or RegionName = 'Piemonte'").groupby('RegionName').sum('TotalPositiveCases')

display(table5)
```

Test 6

Finally we are going to create a final chart to see the recovery rate of the total hospitalized patient over time for the specific regions mentioned above

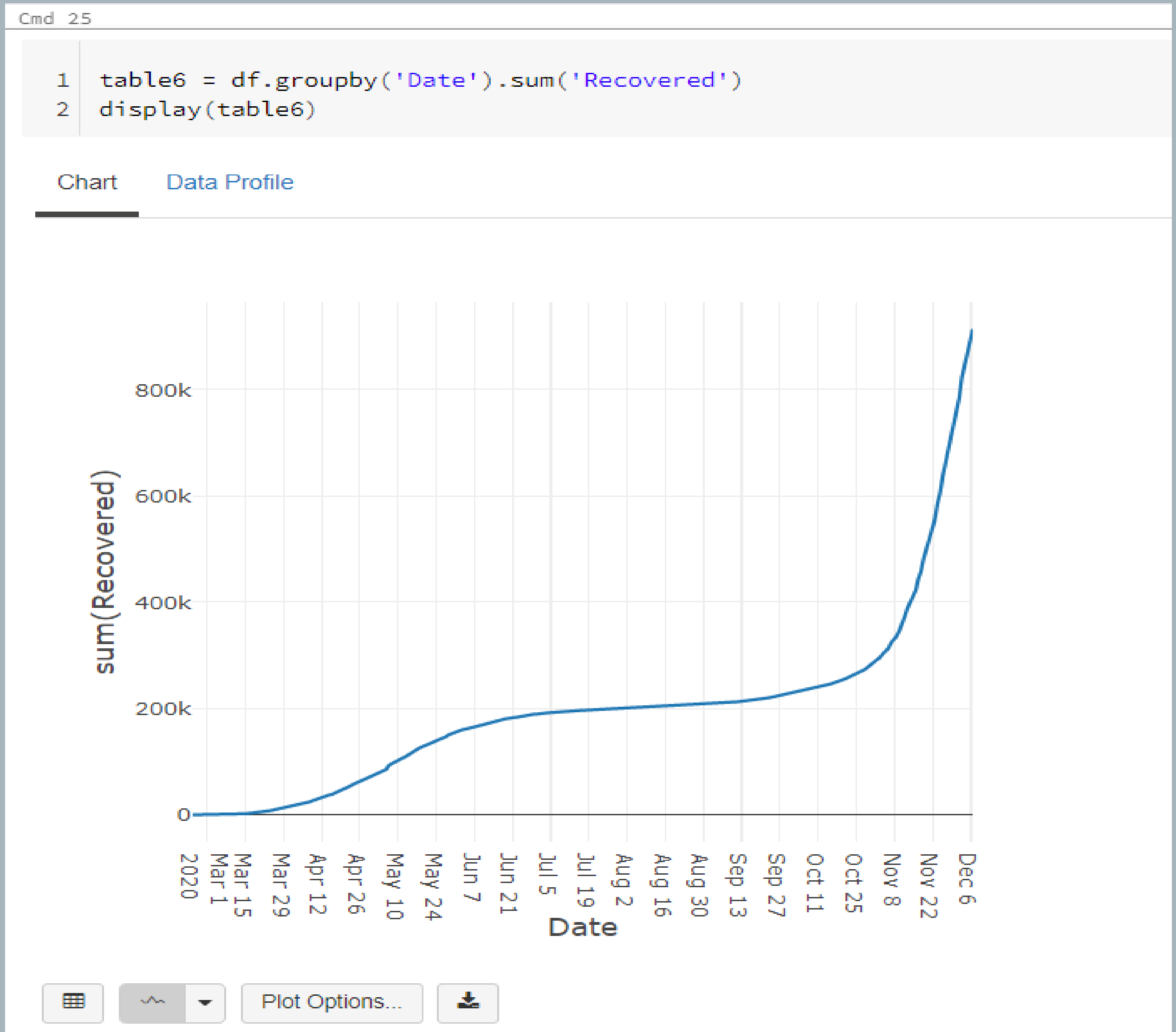


```
table6 = df.filter("RegionName = 'Lombardia' or RegionName = 'Sardegna' or RegionName = 'Piemonte'").groupby('Date').sum('Recovered')
display(table6)
```

	Date	sum(Recovered)
1	2020-06-18T17:00:00.000+0000	87399
2	2020-06-20T17:00:00.000+0000	88663
3	2020-09-15T17:00:00.000+0000	106848
4	2020-09-23T17:00:00.000+0000	108712
5	2020-05-11T17:00:00.000+0000	49156
6	2020-08-22T17:00:00.000+0000	104038
7	2020-11-16T17:00:00.000+0000	208799

Showing all 287 rows.

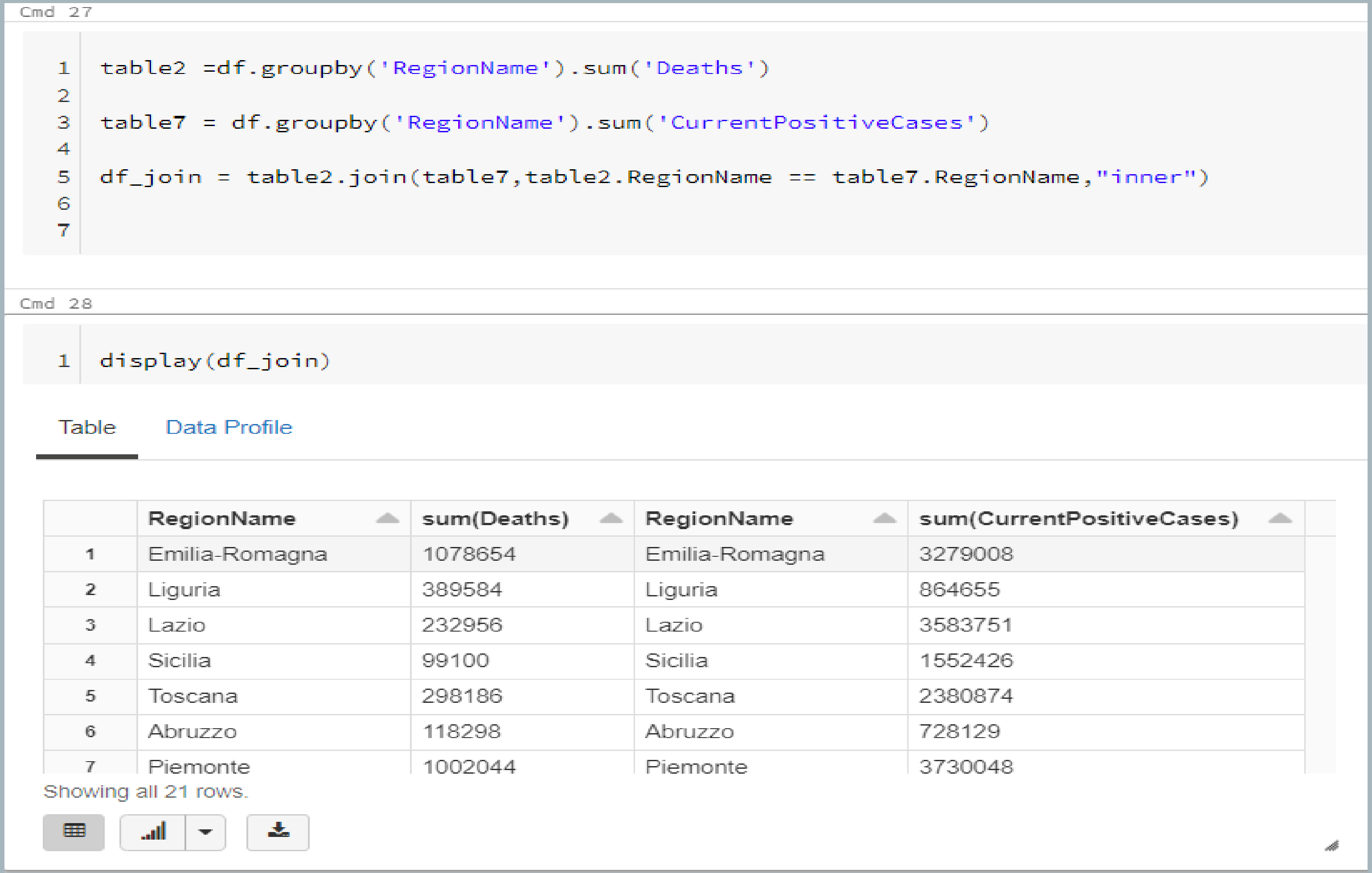
Graph 6



```
table6 = df.groupby('Date').sum('Recovered')
display(table6)
```

Test 7

Now we are going to create another table summing the total patients that were tested positive for each region and join it with the table2, in order to do the comparisons



```
table2 =df.groupby('RegionName').sum('Deaths')

table7 = df.groupby('RegionName').sum('CurrentPositiveCases')

df_join = table2.join(table7,table2.RegionName == table7.RegionName,"inner")
```

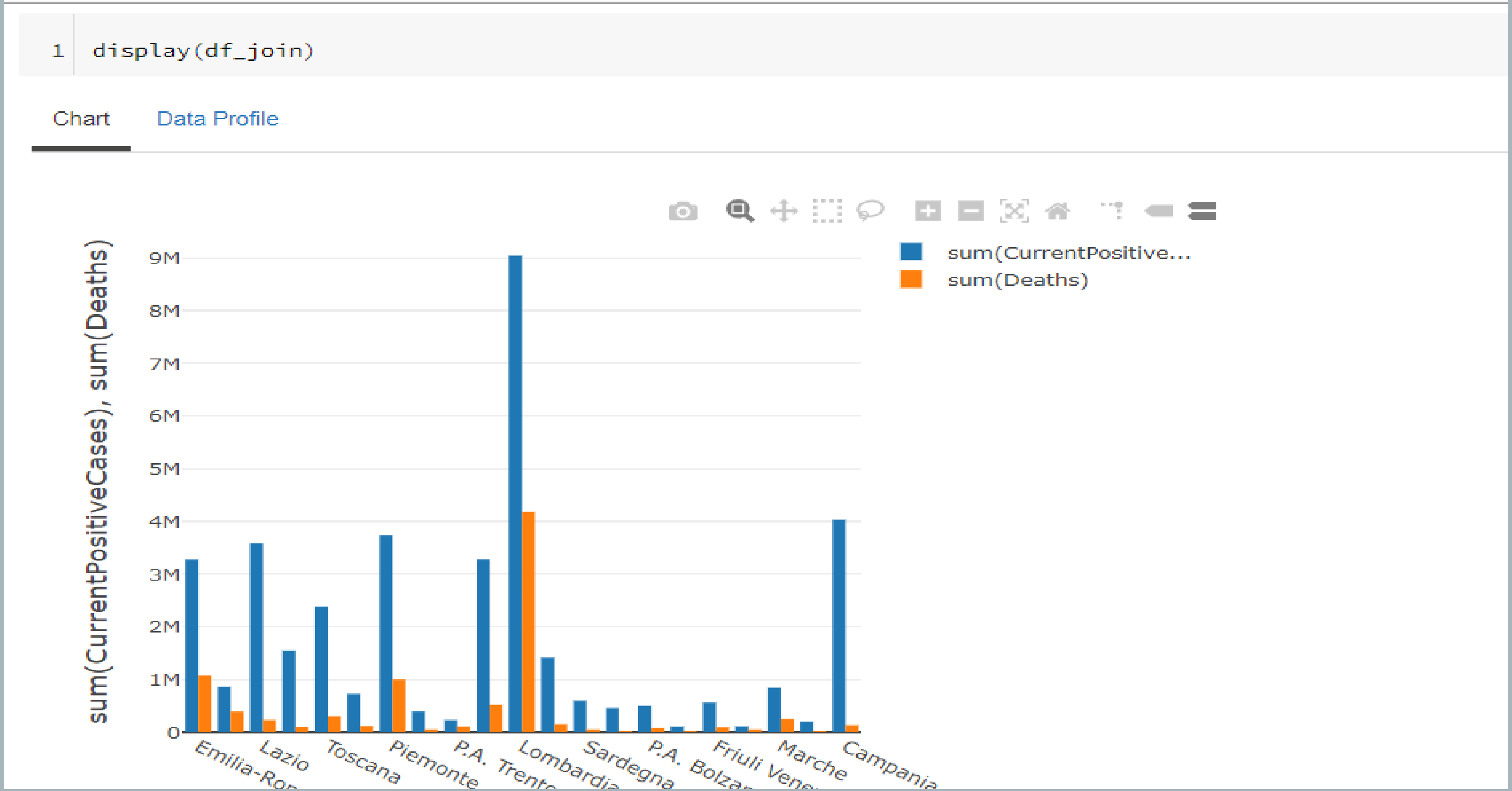
```
display(df_join)
```

	RegionName	sum(Deaths)	RegionName	sum(CurrentPositiveCases)
1	Emilia-Romagna	1078654	Emilia-Romagna	3279008
2	Liguria	389584	Liguria	864655
3	Lazio	232956	Lazio	3583751
4	Sicilia	99100	Sicilia	1552426
5	Toscana	298186	Toscana	2380874
6	Abruzzo	118298	Abruzzo	728129
7	Piemonte	1002044	Piemonte	3730048

Showing all 21 rows.

Graph 7

Graphical representation of df_join table created to display and compare for each region in Italy the sum of the total positive cases compared to the number of total deaths of people you were tested positive.



```
display(df_join)
```

Thank you

Vasilios Philippou
Sofia Theochari