UNIVERSITY OF CENTRAL ARKANSAS

DEPARTMENT OF COMPUTER SCIENCE

CSCI 6385 Topics in Artificial Intelligence (Spring 2016)

FINAL REPORT: CEP (Complex Event Processing)

001189145

SERTAN KAYA

**INFORMATION REGARDING TAXI DATA**

* It consists of 13,798 txt files and its size is 9.4 Gigabyte
* Data format:
  + Name (License plate name)
  + Time
  + jd (Longitude)
  + wd (Latitude)
  + status (0=no passenger, 1=has passenger)
  + V (velocity)
  + Angle (Directions: 0=E, 1=SE, 2=S, 3=SW, 4=W, 5=NW, 6=N, 7=NE )
* Record date: from 2011/04/18 to 2011/04/26 (00:00 to 12:00)

**1st APPROACH: EVENT DETECTION FOR CEP WITH IPYTHON (BACKEND APACHE SPARK) WITH ONLY 125 TXT FILES**

* 125 txt files are used for detecting complex event processing (CEP)
* 1,442,410 row exist in 125 txt files
* 1st Event: Taxi takes passenger (Status=1) Taxi leaves passenger (Status=0)
* Detected Number of Events : 352,968

In this approach, I used IPython (Interface Python) scripting language, in which Apache Spark engine is running in backend. As it is seen in Figure 1, the first thing I did is to ensure Spark Context exist by typing sc command. Then I included essential python libraries such as pandas (python data analysis library ), numpy (numerical python library) and also SQL context. Then I read the 125 txt files which is shown in Line 5 in Figure 1.

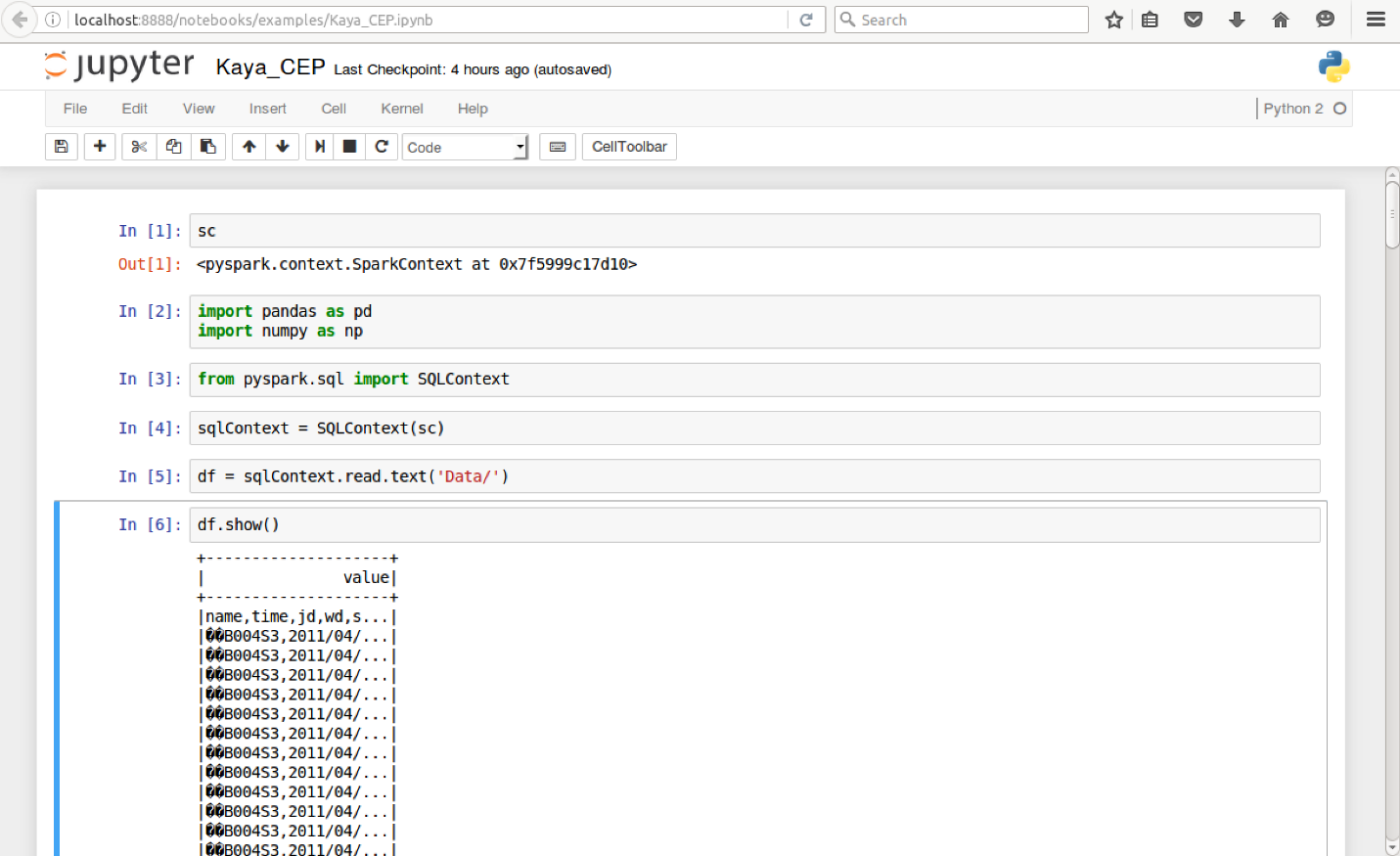
****

Figure 1. Library and SQL context

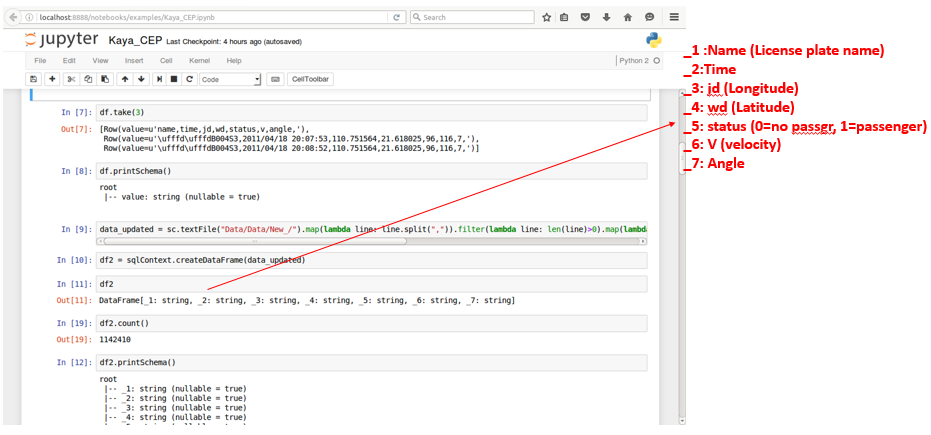


Figure 2. Filtering and creating a data frame

And then, I did mapping and filtering operation on the data. So that, it become a data frame (df) in which each label is converted into columns ranging from \_1 to \_7 numbers. This is also illustrated in Figure 2. As it is shown in Figure 3, the first event, note that status==1 (carry passenger), is detected with the help of command in line 16. I counted number of events which is 352968 and showed some of them in Figure 3.

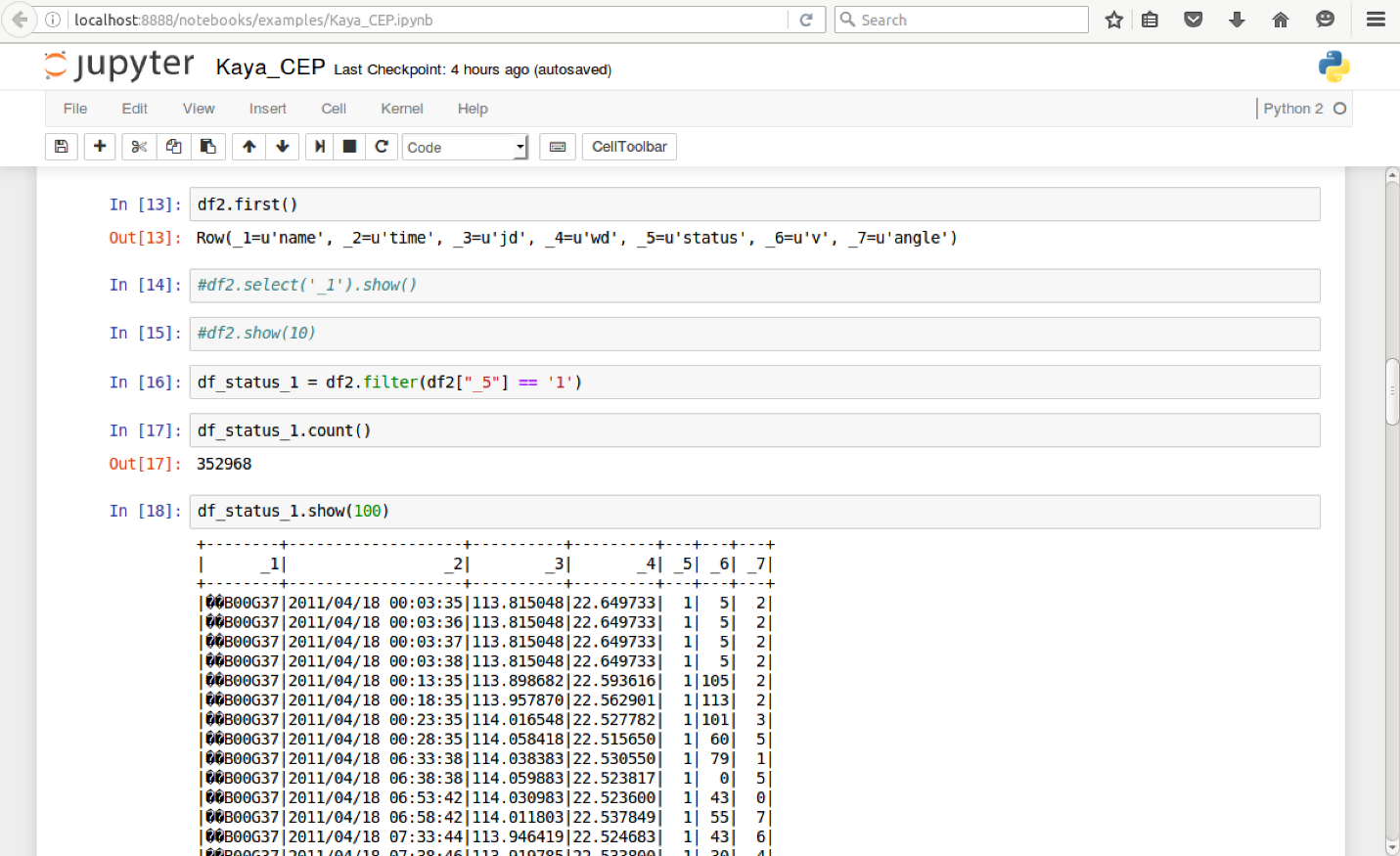


Figure 3. Detecting 1st event

Next, I wanted to detect the taxis which speed more than 0 and carry passenger. For that reason, what I did is to put a filter where velocity > 0 and status == 1. 10 of taxis are shown in Figure 4.

With the same manner, I also wanted to see the taxies velocity >50 and status==1. This is shown in Figure 5. Lastly, I wanted to find number of taxies which head each direction. For that one, I put a filter where \_7 (direction or angle) is set each direction individually and count them. This is also illustrated in Figure 6 and 7.

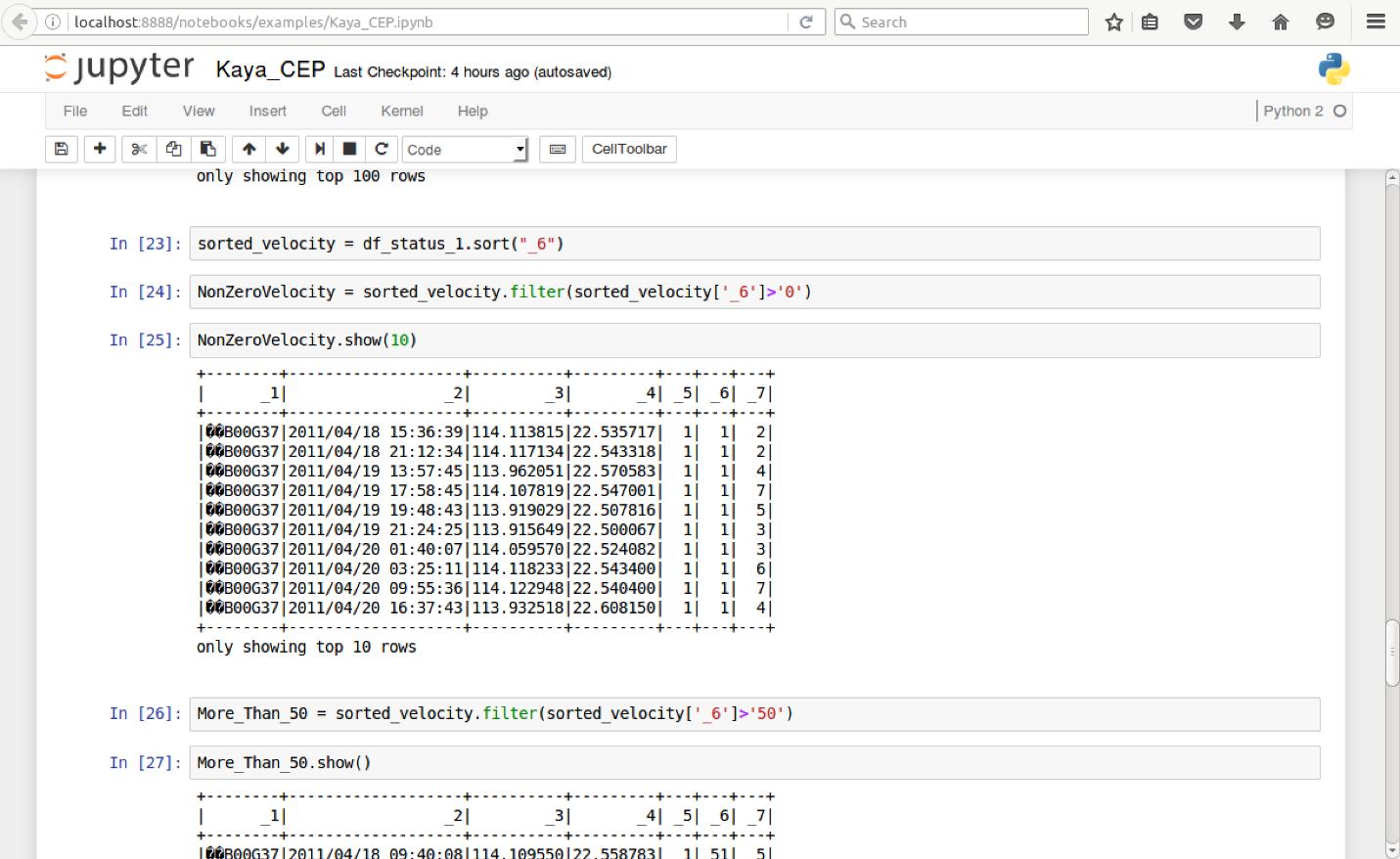


Figure 4. The taxies v>0 and status==1

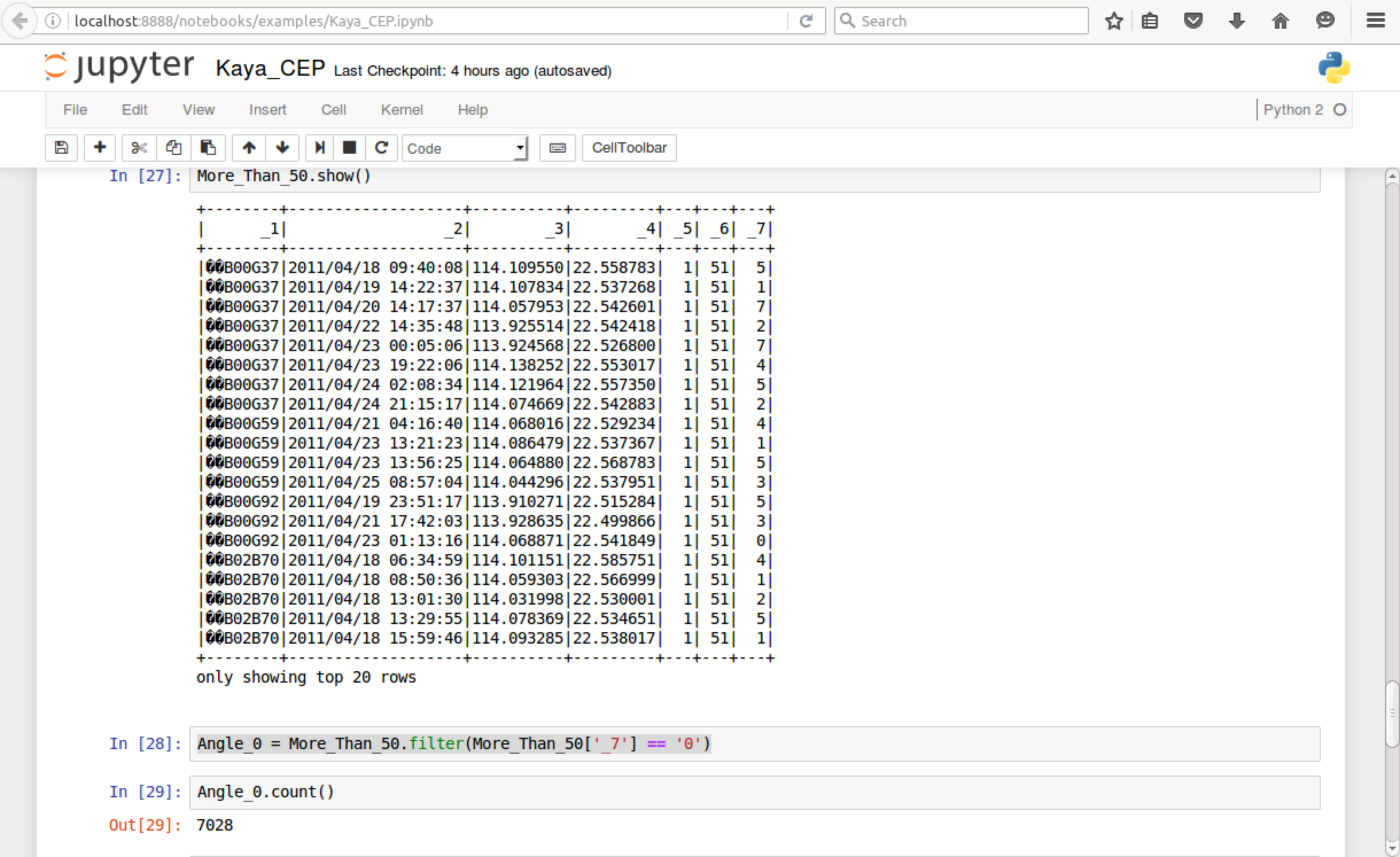


Figure 5. The taxies v>50 and status==1

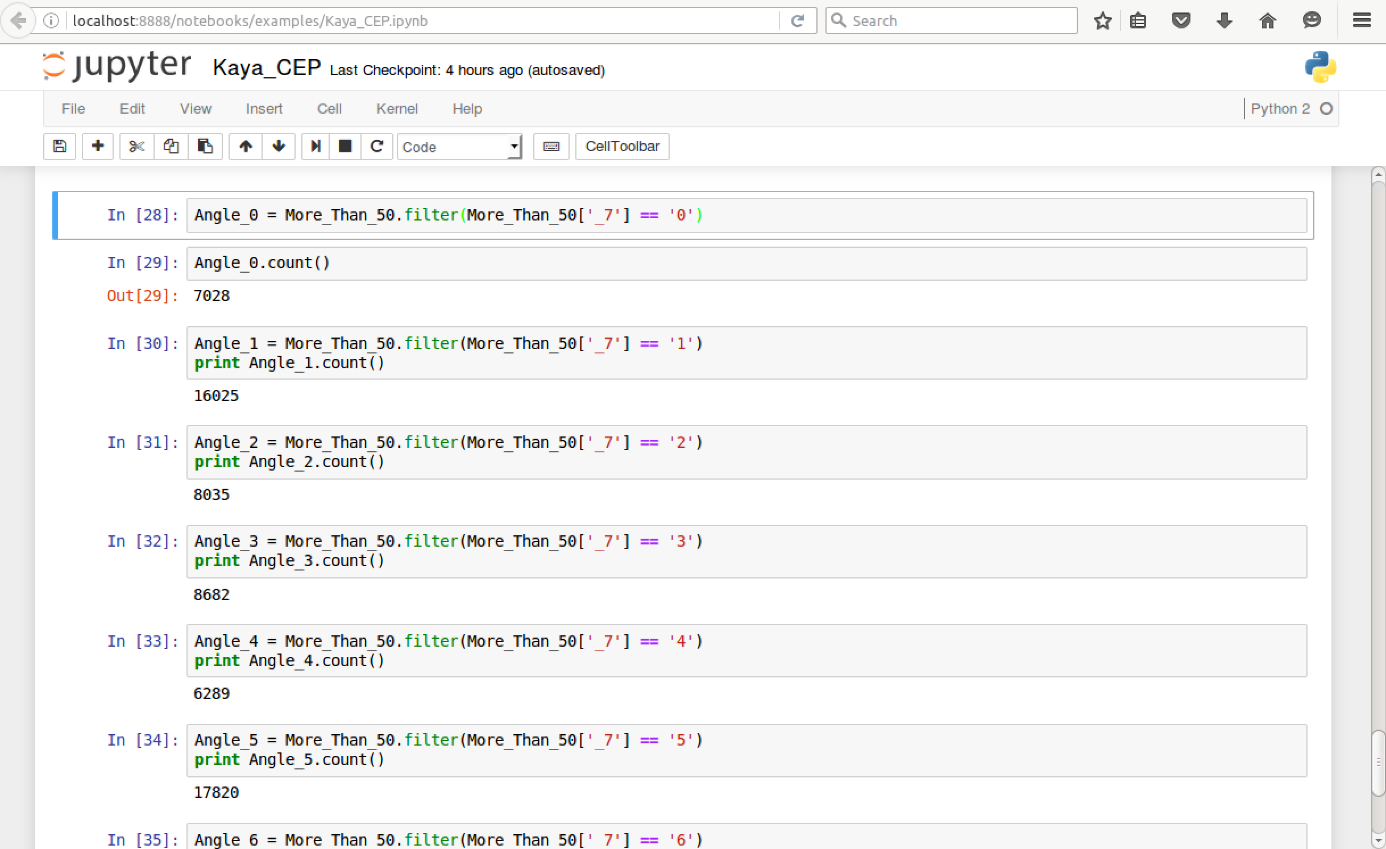


Figure 6. The number of taxies in each direction

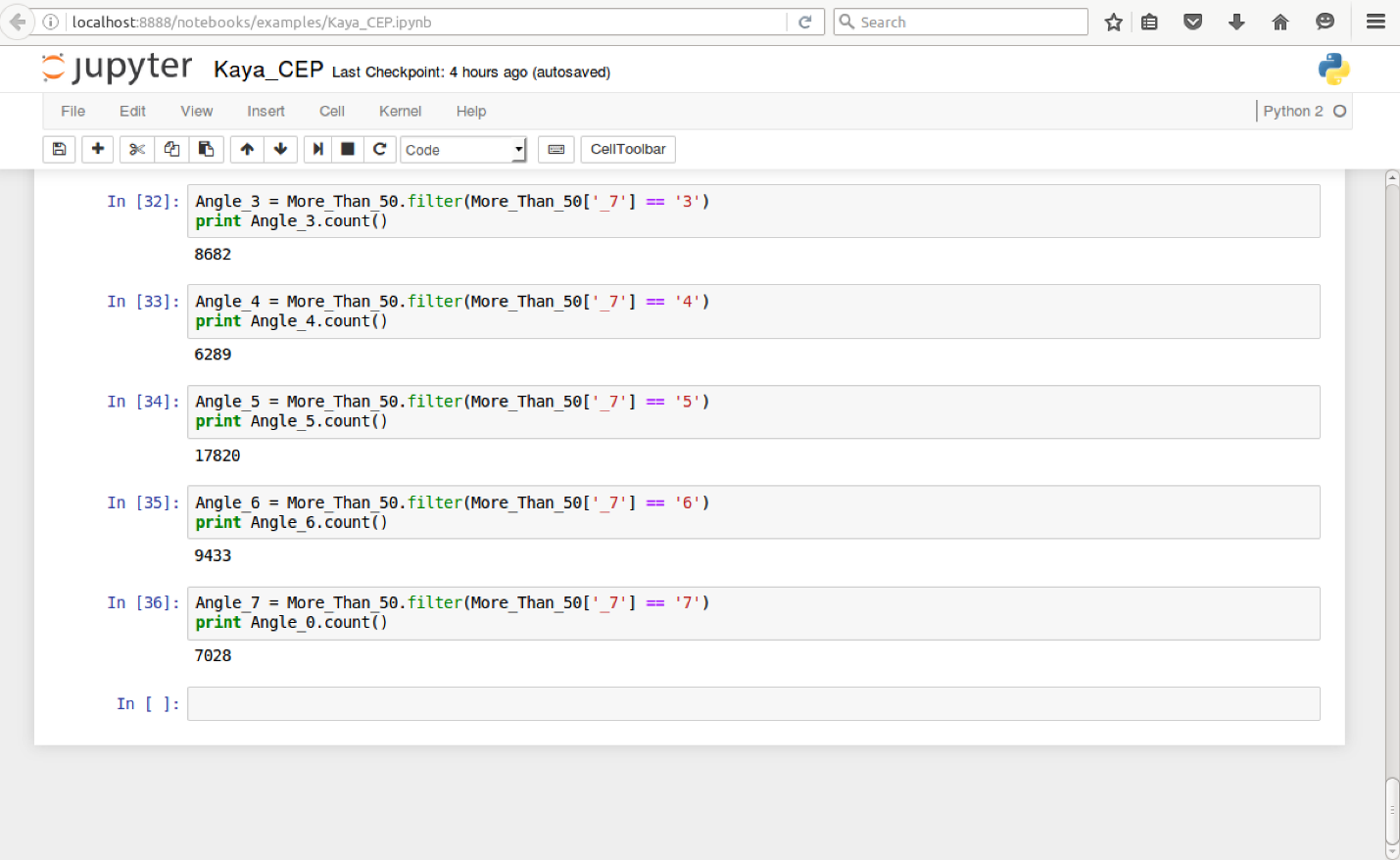
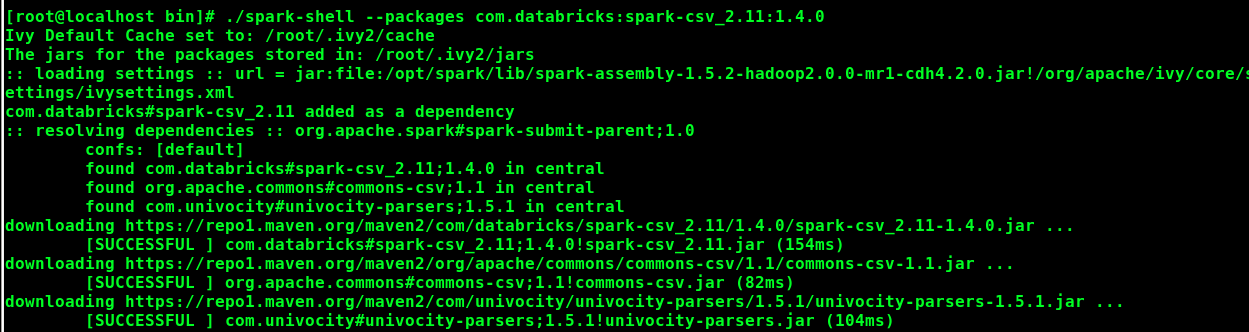


Figure 7. The number of taxies in each direction

**2nd APPROACH: EVENT DETECTION FOR CEP WITH SCALE WITH ENTIRE DATA SET (9.4 GB)**

* 13,798 single txt file is combined into a single taxiData.cvs file
* taxiData.cvs file’s size is 9.4 Gigabyte
* 1st Event: Taxi takes passenger (Status=1) Taxi leaves passenger (Status=0)

First, I run spark-shell including essential libraries which is shown in Figure 8.



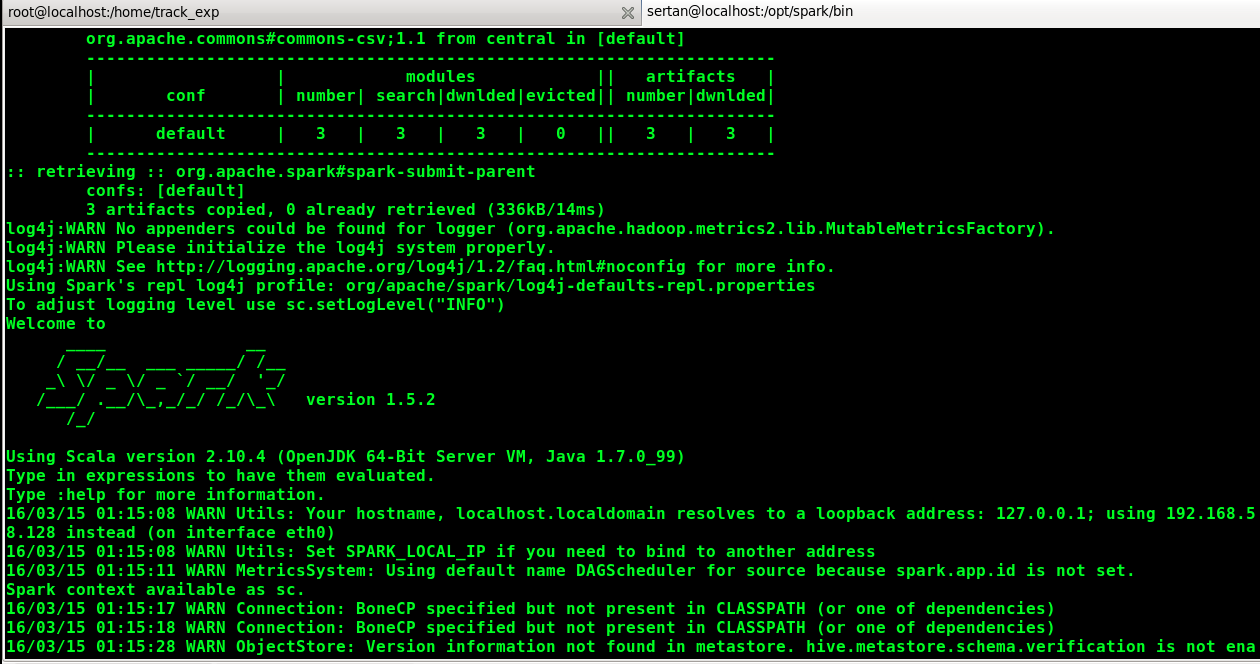


Figure 8. Running spark shell calling particular library

Then, I imported several libraries and define SQL context using scala scripting language. This is illustrated in Figure 9.

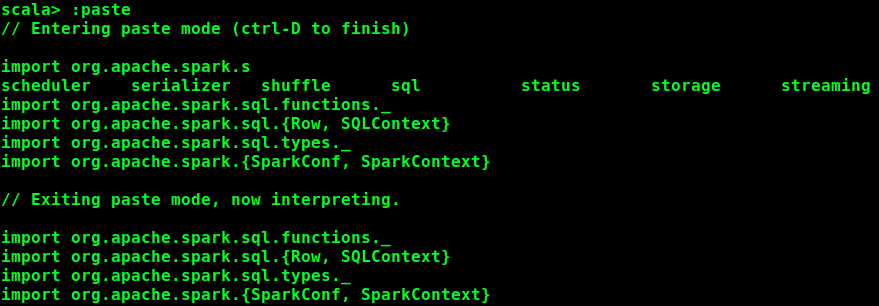




Figure 9. Including libraries and defining SQL context

Then, I read, formated and showed only 20 rows of the taxiData which is shown in Figure 10.

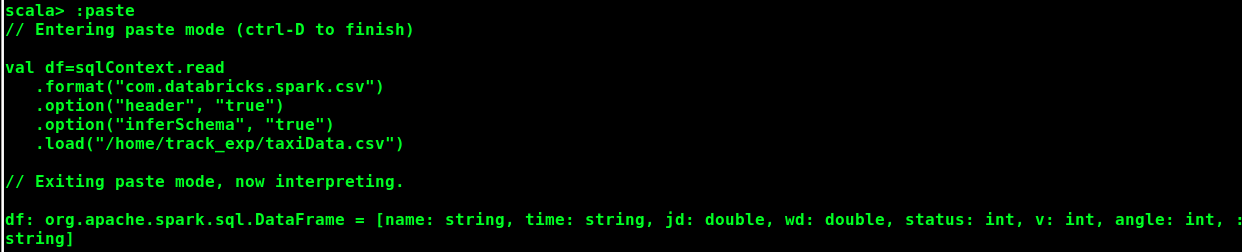
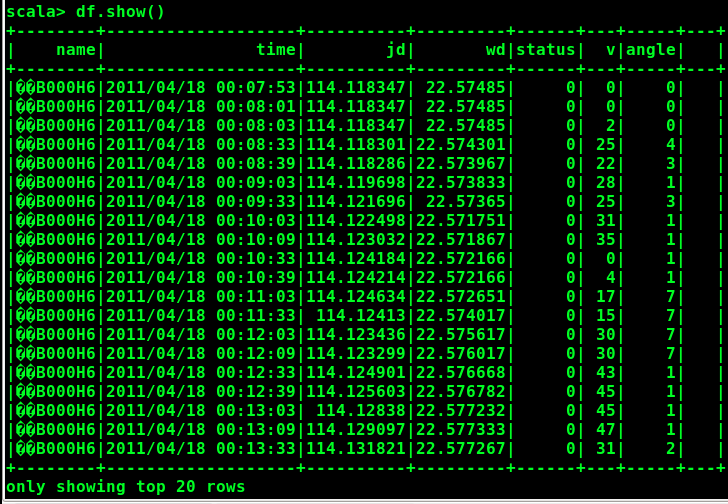


Figure 10. Reading, formatting and showing of the data

Then, I grouped by license plate, counted them and showed top 20 of them as seen in Figure 11.

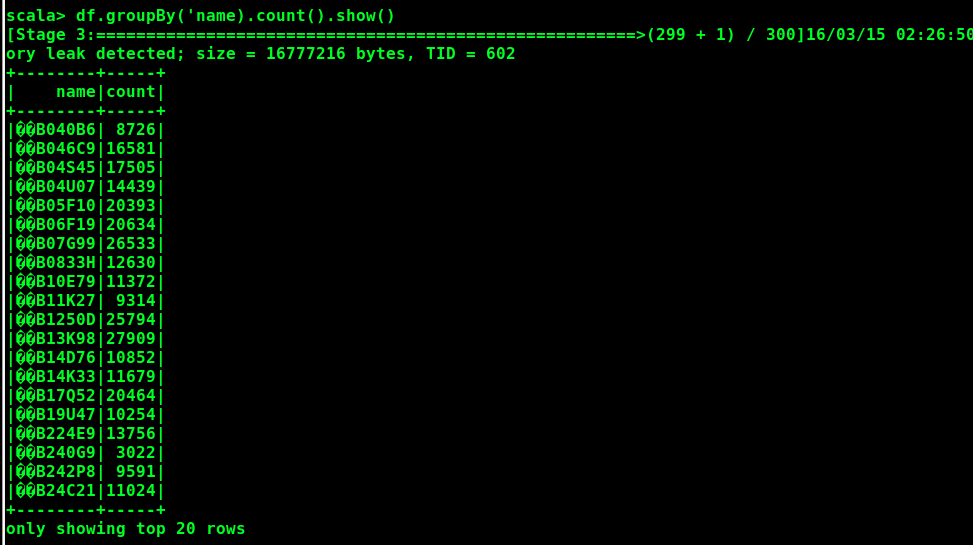


Figure 11. Reading, formatting and showing of the data

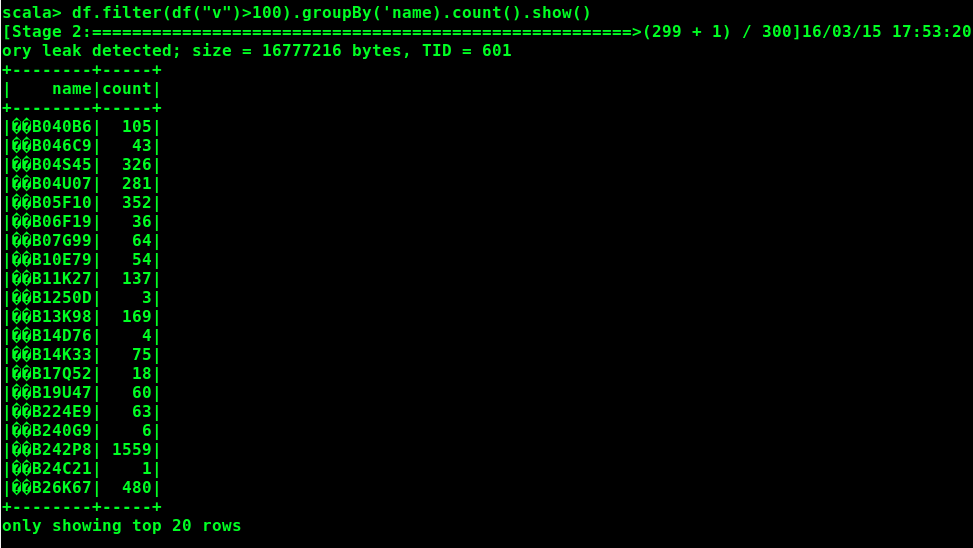
Then, I wanted to detect the taxies which speed at more 100 and count these. This is shown in Figure 12.

Figure 12. The taxies > 100 and counted of them

After that, I wanted to find out average velocity of the taxies while carrying passenger. This event is shown in Figure 13.

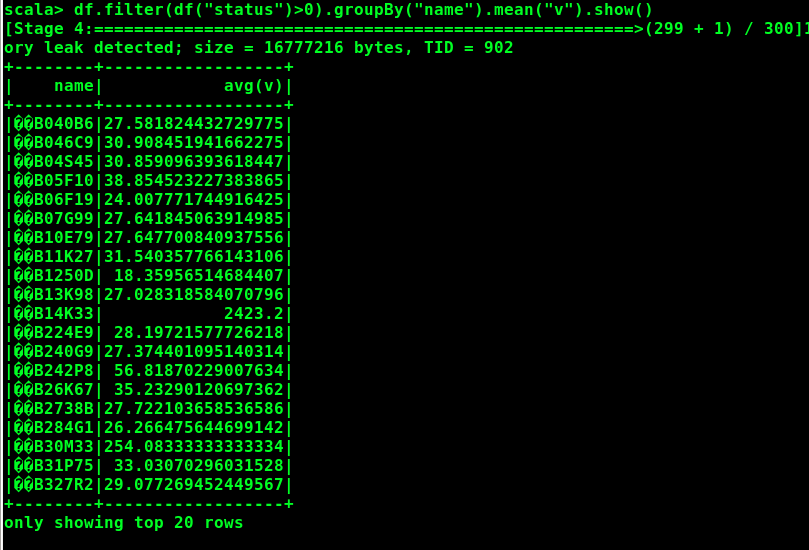


Figure 13. The taxies average speed while carrying passenger

I also counted taxies while carrying passenger as it is shown in Figure 14.

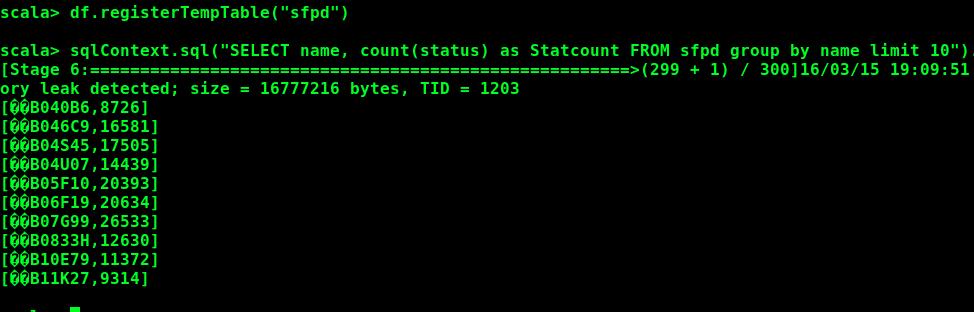


Figure 14. Counting the taxies while carrying passenger

Figure 15 shows that the average speed of taxies and limiting only 10 taxies.

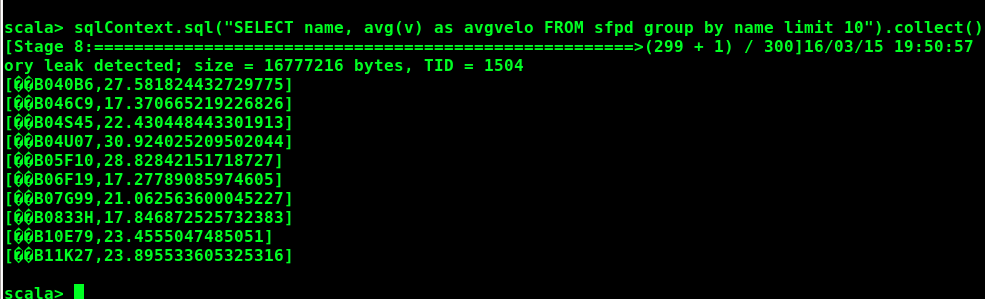


Figure 15. Average speed of the taxies

In Figure 16, maximum velocities of taxies are shown while not carrying any passenger.

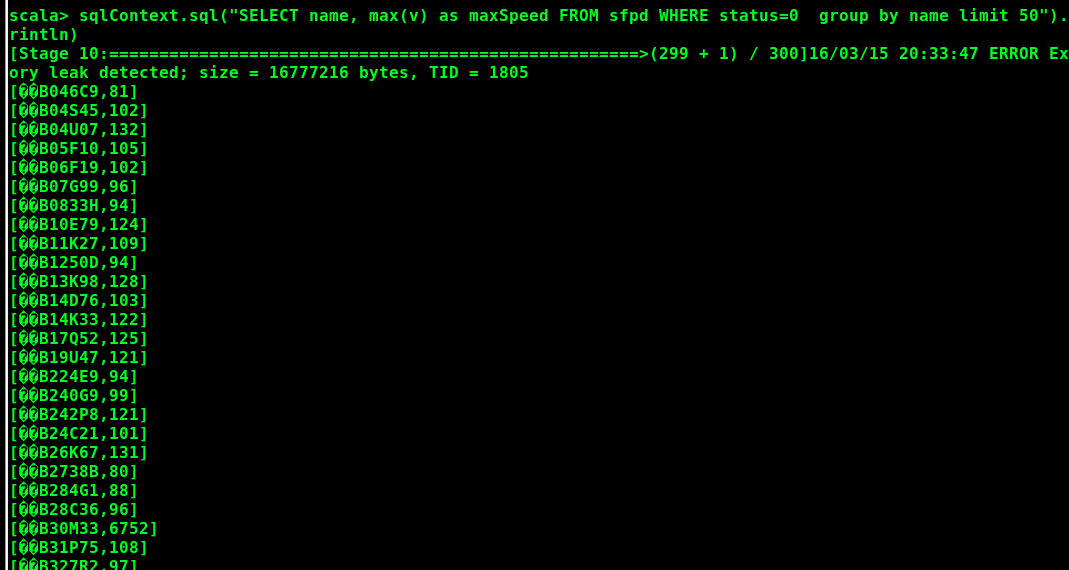


Figure 16. Maximum velocities of the taxies while not carrying passenger

Finally, in Figure 17 maximum velocities of taxies are shown while carrying any passenger. That means that status ==1 and find max v.

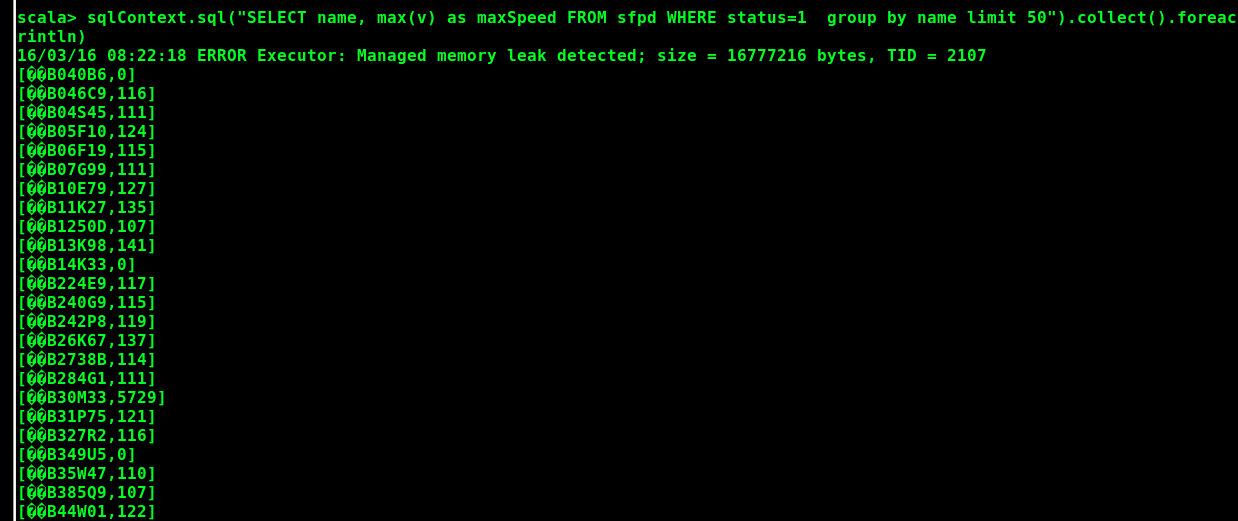


Figure 17. Maximum velocities of the taxies while carrying passenger

**GRAPH CHART FOR 2ND APPROACH VIA TABLEAU TOOL**

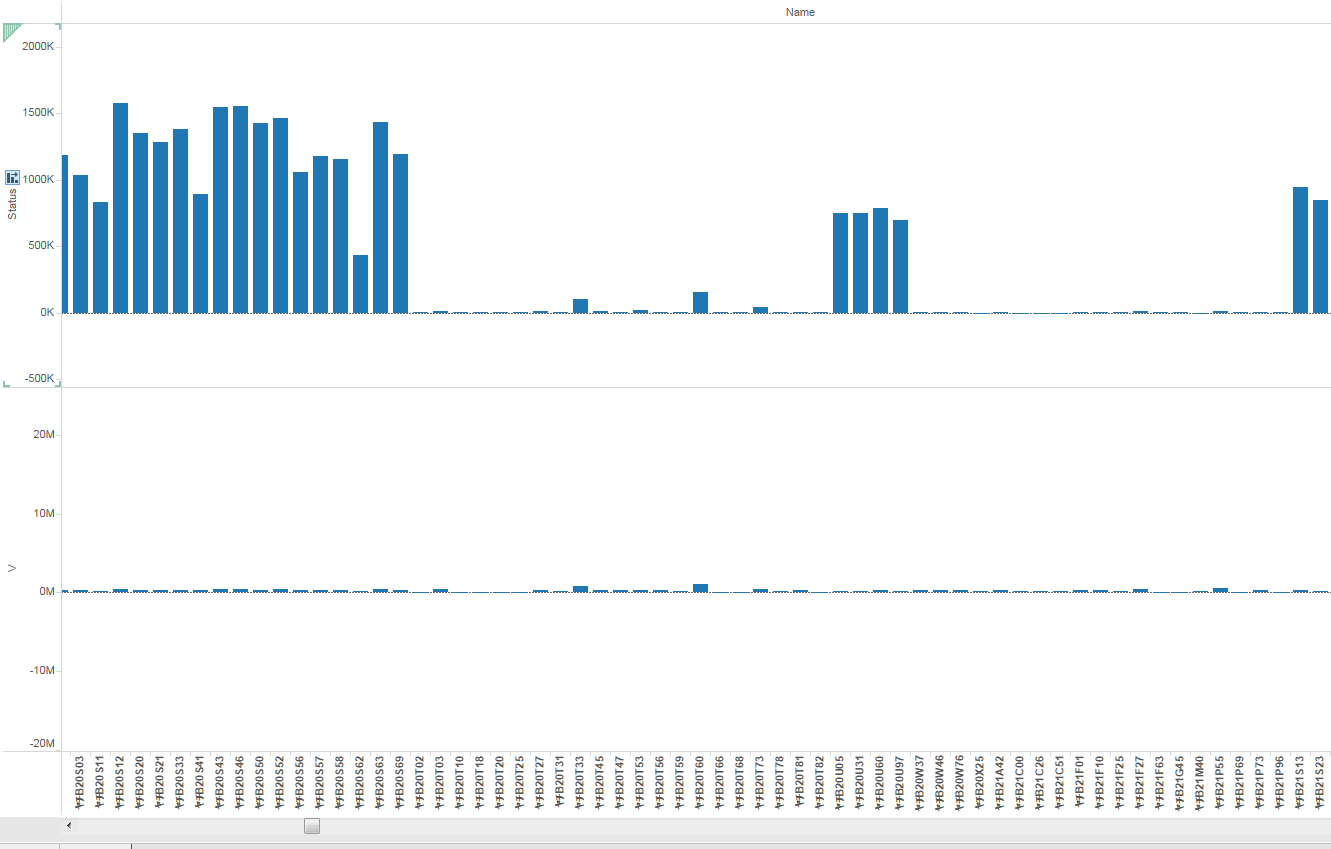


Figure 18. Bar chart for the entire data set (License plate by V and Status)

In this section, I used the Tableau tool to visualize and interpret the taxiData set. The first graph, which is seen in Figure 18, is the bar chart. In this graph, the horizontal axis (x axis) shows the license plate name of taxies and the vertical axis (y axis) shows v velocity on the bottom part and status in the upper side of the graph. As it is seen the graph, several cars have high bars which means they have many passengers in this time frame. The status basically shows total number of the passenger for each taxi. In the velocity section, we see the sum of the velocities for each taxi. When we want to see all oscillation on Figure 19. As you can notice that, there are some noises in the data. Basically, the taxi data includes some noises on it. In order to better analyze the data, the noisy data should be excluded so that the CEP results could be more accurate and prices.

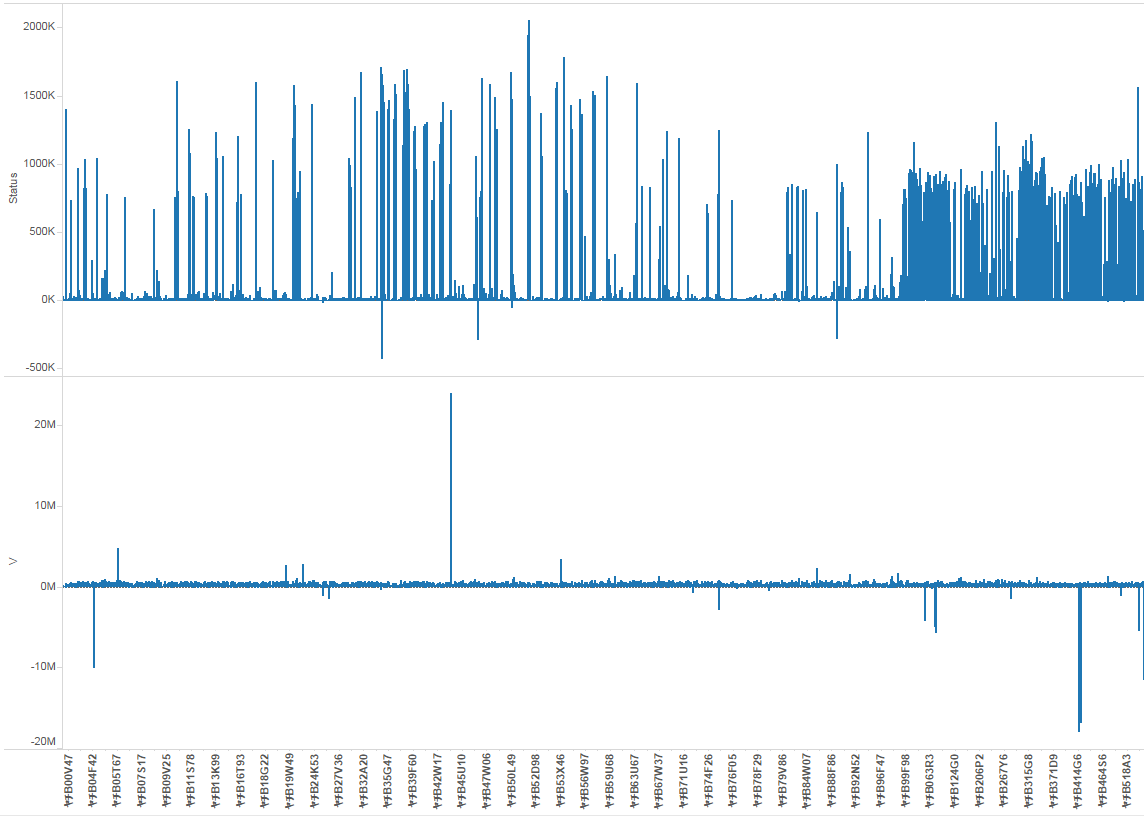


Figure 19. License plate by V and Status with noisy

Finally, in order to do better interpretation, I draw a circle graph which is shown in Figure 20. This graph tells us that which taxi made more changing in direction. Therefore, we can conclude that the taxi goes through roads which included variety of directions such as left, right, up and down.

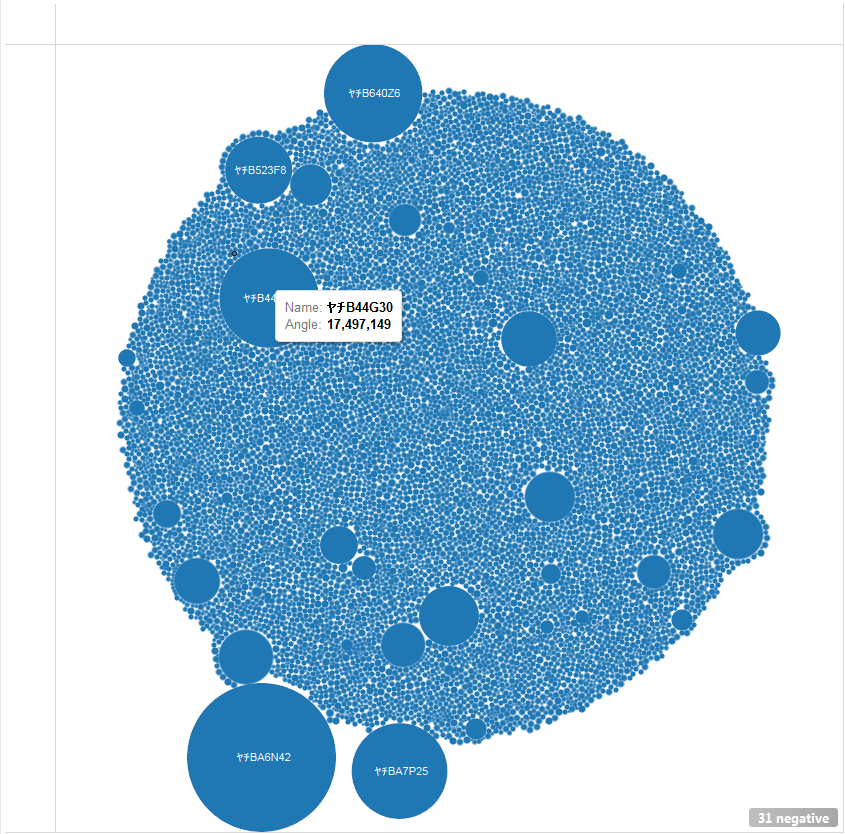


Figure 20. Circle graph based upon angle (direction)