

# **CS5540 - Principles of Big Data Management**

## **Project Phase-2**

### **Twitter Data Analysis of Cricket Premier Leagues**

**By**

Yagnasri Chowdary Nalluri - 16293464

Akarsha Yedla - 16293464

Pravallika Reddy Lankapothu- 16293464

## Objective:

This part of project deals with Analysis of created data and also its visualization based on various parameters. Here we come with the queries as mentioned in the interim report.

## Technologies used:

- JetBrains PyCharm 2018.3.5
- Spark
- Scala
- Tableau 2019.4

## Queries and Analysis:

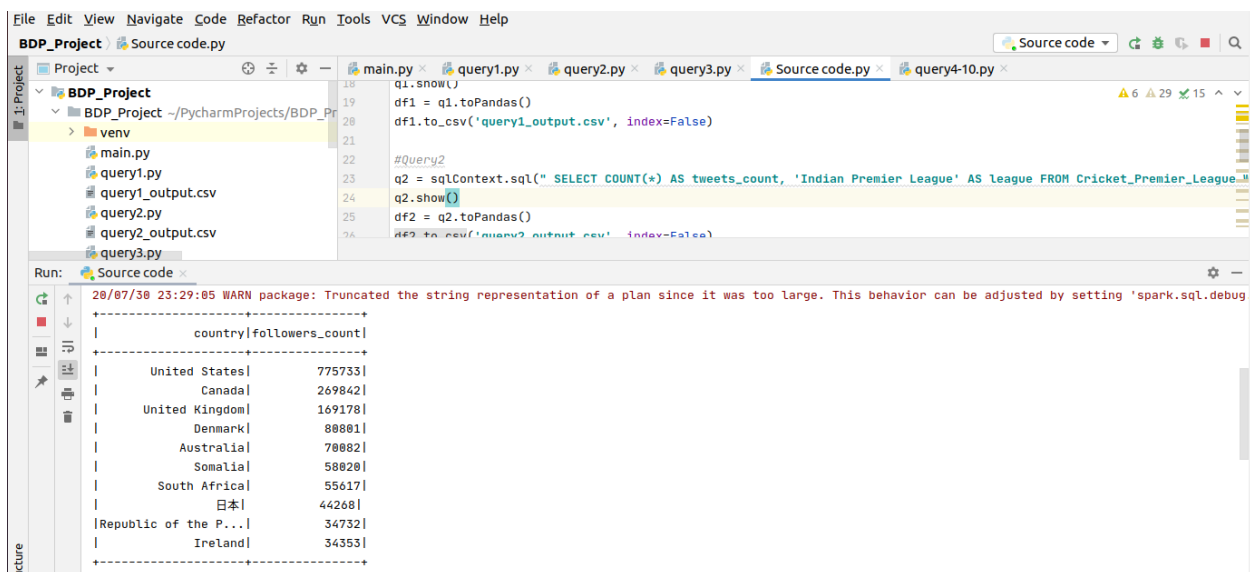
### Query 1:

This query collects the top 10 geographical locations from which highest number of tweets had been recorded.

Please find the query below that had been used for extraction

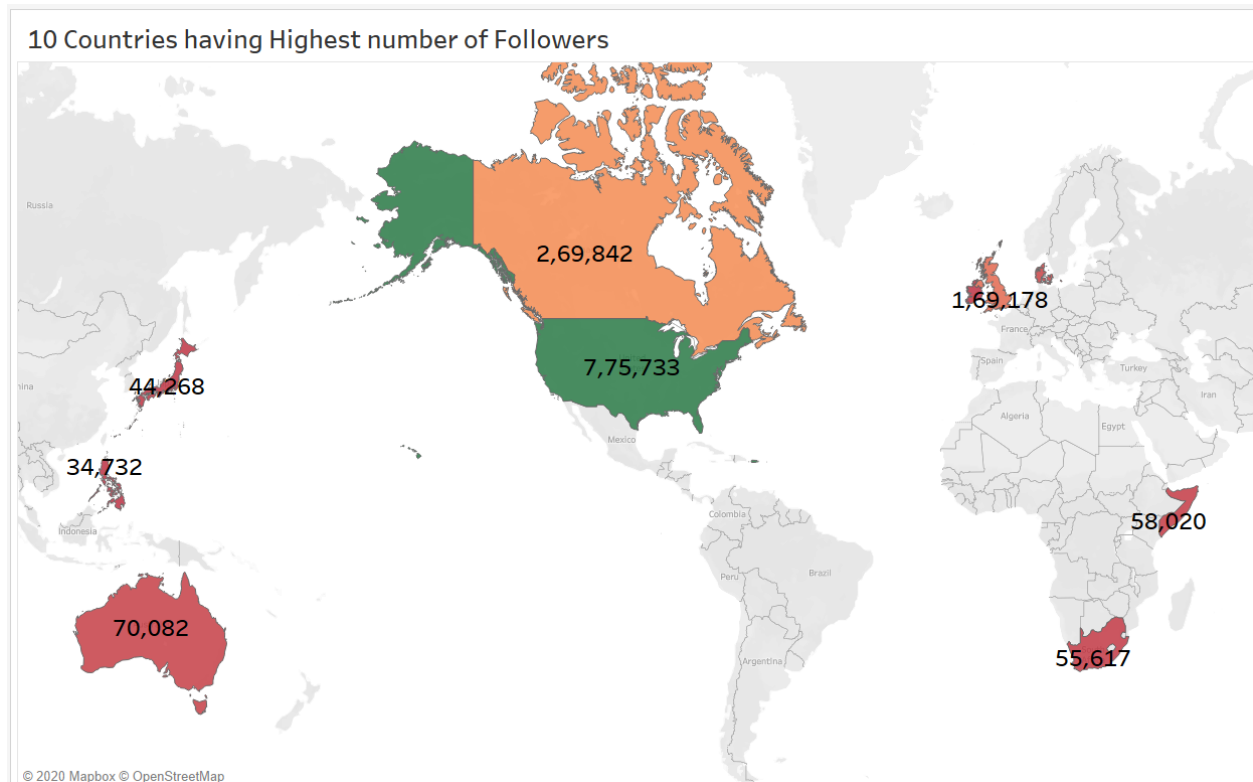
```
q1 = sqlContext.sql("SELECT place.country AS country,  
SUM(user.followers_count) AS followers_count FROM Cricket_Premier_League  
WHERE place.country != 'null' GROUP BY place.country ORDER BY  
followers_count DESC LIMIT 10")
```

Please find the screenshot of the query output.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
BDP_Project Source code.py Source code 6 29 15 Q
Project BDP_Project
  BDP_Project
    venv
    main.py
    query1.py
    query1_output.csv
    query2.py
    query2_output.csv
    query3.py
main.py query1.py query2.py query3.py Source code.py query4-10.py
18 q1.show()
19 df1 = q1.toPandas()
20 df1.to_csv('query1_output.csv', index=False)
21
22 #Query2
23 q2 = sqlContext.sql(" SELECT COUNT(*) AS tweets_count, 'Indian Premier League' AS league FROM Cricket_Premier_League_M
24 q2.show()
25 df2 = q2.toPandas()
26 df2.to_csv('query2_output.csv', index=False)
Run: Source code
28/07/30 25:29:05 WARN package: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug
country|followers_count|
+-----+-----+
| United States| 775733|
| Canada| 269842|
| United Kingdom| 169178|
| Denmark| 88881|
| Australia| 70882|
| Somalia| 58828|
| South Africa| 55617|
| 日本| 44268|
| Republic of the P...| 34732|
| Ireland| 34353|
+-----+-----+
```

Please find the screenshot of the data visualization for query 1.



## Query 2:

This query is used to collect the number of recorded for the leagues “IPL”, “PSL”, “CPL”, “BPL” and “Ashes” and sort them in the descending order.

Please find the query below that had been used for extraction

```
q2 = sqlContext.sql(" SELECT COUNT(*) AS tweets_count, 'Indian Premier League' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%ipl%' OR TEXT LIKE '%Indian Premier League%' OR TEXT LIKE '%IPL%' OR TEXT LIKE '%Ipl%' UNION SELECT COUNT(*) AS tweets_count, 'Pakistan Super League' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%psl%' OR TEXT LIKE '%Pakistan Super League%' OR TEXT LIKE '%PSL%' OR TEXT LIKE '%Psl%' UNION SELECT COUNT(*) AS tweets_count, 'Caribbean Premier League' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%cpl%' OR TEXT LIKE '%Caribbean Premier League%' OR TEXT LIKE '%CPL%' OR TEXT LIKE '%Cpl%' UNION SELECT COUNT(*) AS tweets_count, 'Bangladesh Premier League' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%bpl%' OR TEXT LIKE '%Bangladesh Premier League%' OR TEXT LIKE '%BPL%' OR TEXT LIKE '%Bpl%' UNION SELECT COUNT(*) AS tweets_count, 'Ashes' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%ashes%' OR TEXT LIKE '%Ashes%' OR TEXT LIKE '%ASHES%' UNION SELECT COUNT(*) AS tweets_count, 'Big Bash League' AS league FROM Cricket_Premier_League WHERE TEXT LIKE '%bbl%' OR TEXT LIKE '%Big Bash League%' OR TEXT LIKE '%BBL%' OR TEXT LIKE '%Bbl%' OR TEXT LIKE '%big bash%'")
```

Please find the screenshot of the query output.

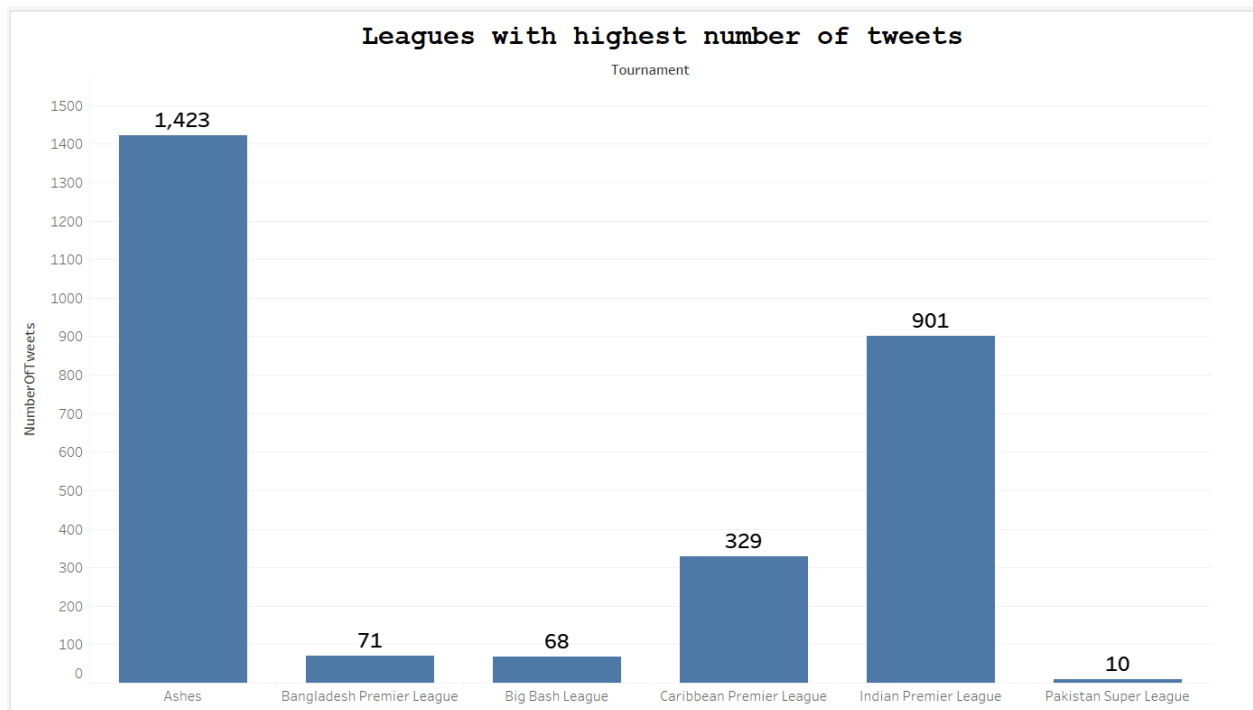


```
18 q1.show()
19 df1 = q1.toPandas()
20 df1.to_csv('query1_output.csv', index=False)
21
22 #Query2
23 q2 = sqlContext.sql(" SELECT COUNT(*) AS tweets_count, 'Indian Premier League' AS League FROM Cricket_Premier_League")
24 q2.show()
25 df2 = q2.toPandas()
26 df2.to_csv('query2_output.csv', index=False)
```

Run: Source code

League	tweets_count
Republic of the P...	34732
Ireland	34353
-----	
tweets_count	League
-----	
71	Bangladesh Premie...
901	Indian Premier Le...
329	Caribbean Premier...
1423	Ashes
10	Pakistan Super Le...
68	Big Bash League

Please find the screenshot of the data visualization for query 2.



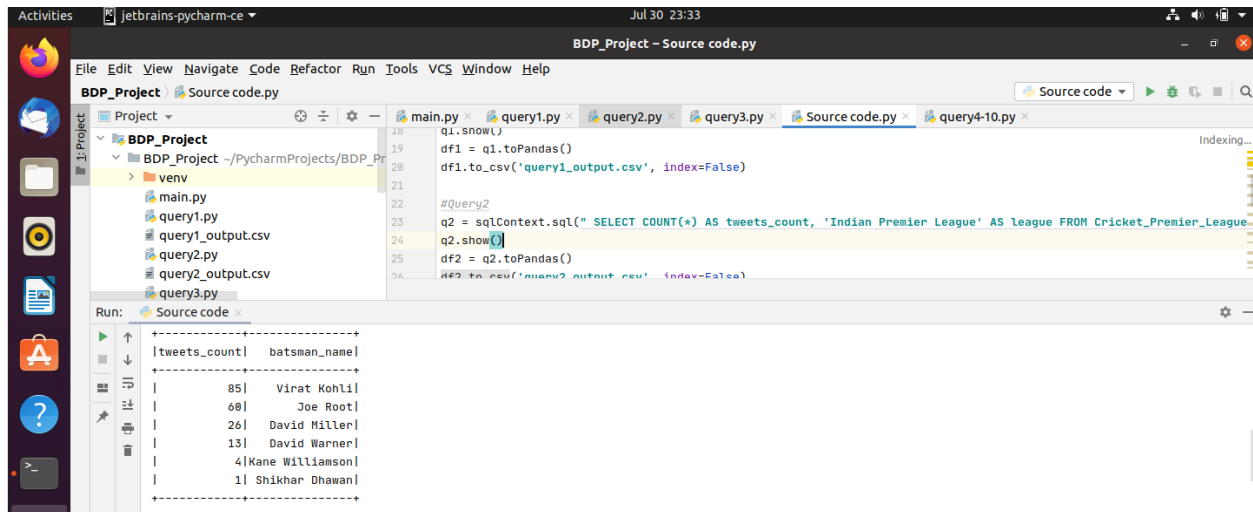
### Query 3:

This query is to find the most famous batsman among “Virat Kohli”, “David Warner”, “Kane Williamson”, “Joe Root”, “Shikhar Dhawan” and “David Miller”.

Please find the query below that had been used for extraction

```
q3 = sqlContext.sql(" SELECT COUNT(*) AS tweets_count, 'Virat Kohli' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%virat kohli%' OR TEXT LIKE '%Virat Kohli%' OR TEXT LIKE '%virat%' OR TEXT LIKE '%kohli%' UNION SELECT COUNT(*) AS tweets_count, 'David Warner' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%david warner%' OR TEXT LIKE '%David Warner%' OR TEXT LIKE '%warner%' OR TEXT LIKE '%WARNER%' UNION SELECT COUNT(*) AS tweets_count, 'Kane Williamson' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%Kane Williamson%' OR TEXT LIKE '%kane williamson%' OR TEXT LIKE '%kane%' OR TEXT LIKE '%williamson%' UNION SELECT COUNT(*) AS tweets_count, 'Joe Root' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%Joe Root%' OR TEXT LIKE '%joe root%' OR TEXT LIKE '%root%' OR TEXT LIKE '%Root%' UNION SELECT COUNT(*) AS tweets_count, 'Shikhar Dhawan' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%Shikhar Dhawan%' OR TEXT LIKE '%shikhar dhawan%' OR TEXT LIKE '%shikhar%' OR TEXT LIKE '%dhawan%' OR TEXT LIKE '%gabbar%' UNION SELECT COUNT(*) AS tweets_count, 'David Miller' AS batsman_name FROM Cricket_Premier_League WHERE TEXT LIKE '%David Miller%' OR TEXT LIKE '%david miller%' OR TEXT LIKE '%Miller%' OR TEXT LIKE '%miller%' OR TEXT LIKE '%Miller%' ORDER BY tweets_count DESC")
```

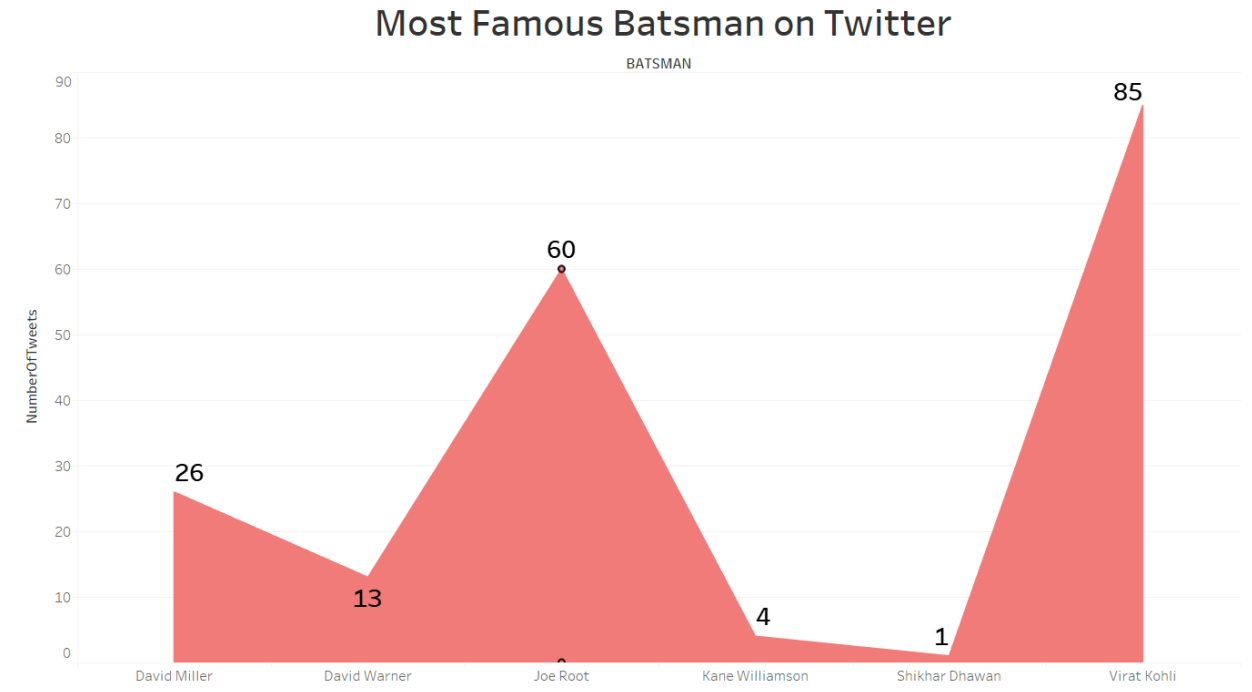
Please find the screenshot of the query output.



The screenshot shows the PyCharm IDE interface. The top toolbar indicates the current file is 'Source code.py'. The main editor window displays a SQL query that counts tweets for various batsmen. The output of the query is shown in a table below the code.

tweets_count	batsman_name
85	Virat Kohli
60	Joe Root
26	David Miller
13	David Warner
4	Kane Williamson
1	Shikhar Dhawan

Please find the screenshot of the data visualization for query 3.



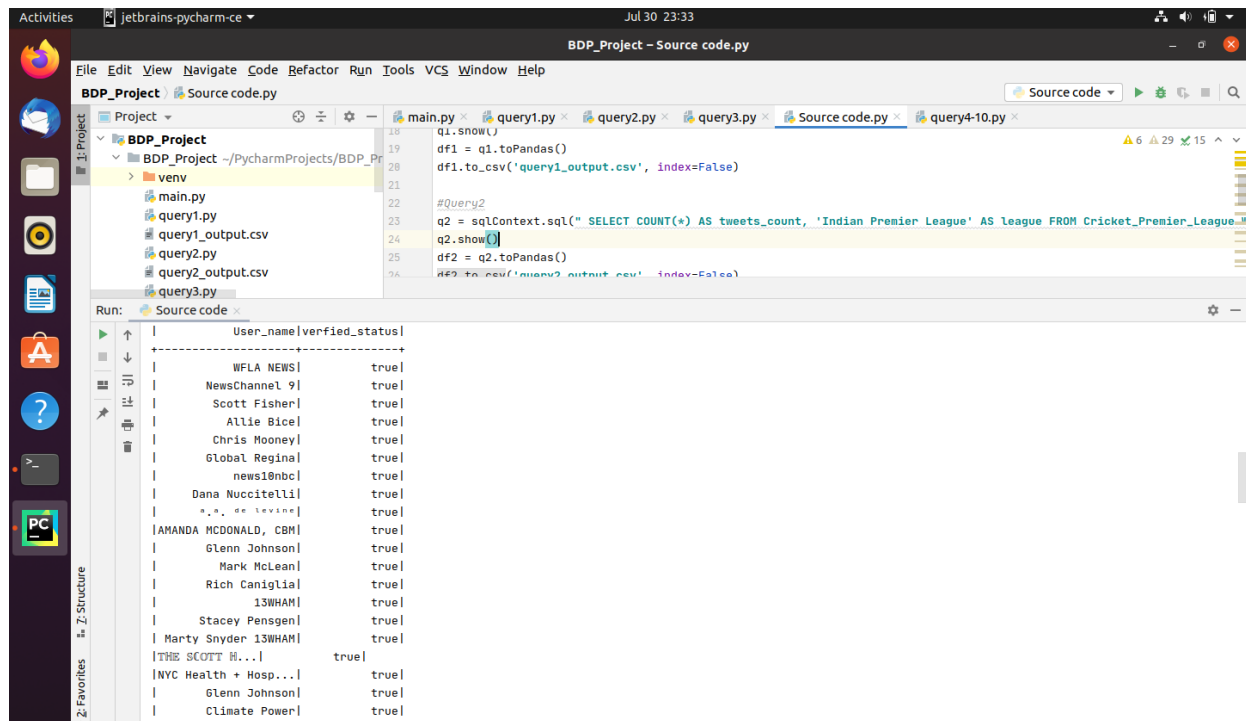
#### Query 4:

This query is used to extract the list of verified users.

Please find the query below that had been used for extraction

```
q4 = sqlContext.sql("SELECT user.name AS User_name, user.verified as  
verified_status FROM Cricket_Premier_League WHERE user.verified = 'true'")
```

Please find the screenshot of the query output.



Please find the screenshot of the data visualization for query 4.

### List of Verified Users

NAME_OF_USER	VERIFICATION_STATUS
4WARN Weather	True
7 Eyewitness News	True
7 Weather	True
7News Boston WHDH	True
8News WRIC Richmo..	True
9NEWS Denver	True
9NEWS Weather	True
10 Tampa Bay	True
10TV	True
12 News	True
13 On Your Side	True
13 Weather Authority	True
13WHAM	True
22News StormTeam	True
95.5 KLOSFM	True
106.3 WORD	True
305 Mosquito Control	True
350 dot org	True
511 Alberta	True
511Ontario	True
680 CJOB	True
770 CHQR Global Ne..	True
935 KDAY	True
@dispatch_DD	True
@himanshu	True
#BLM Mark B Donica	True
#CoastSafe	True
#MaskUpMelbourne	True
+SocialGood	True
🌱 Jesse 🌱 itka 🌱	True
👤 Emily Byrd 📷	True
👤 rosanna arquet..	True
a,a, de levine	True

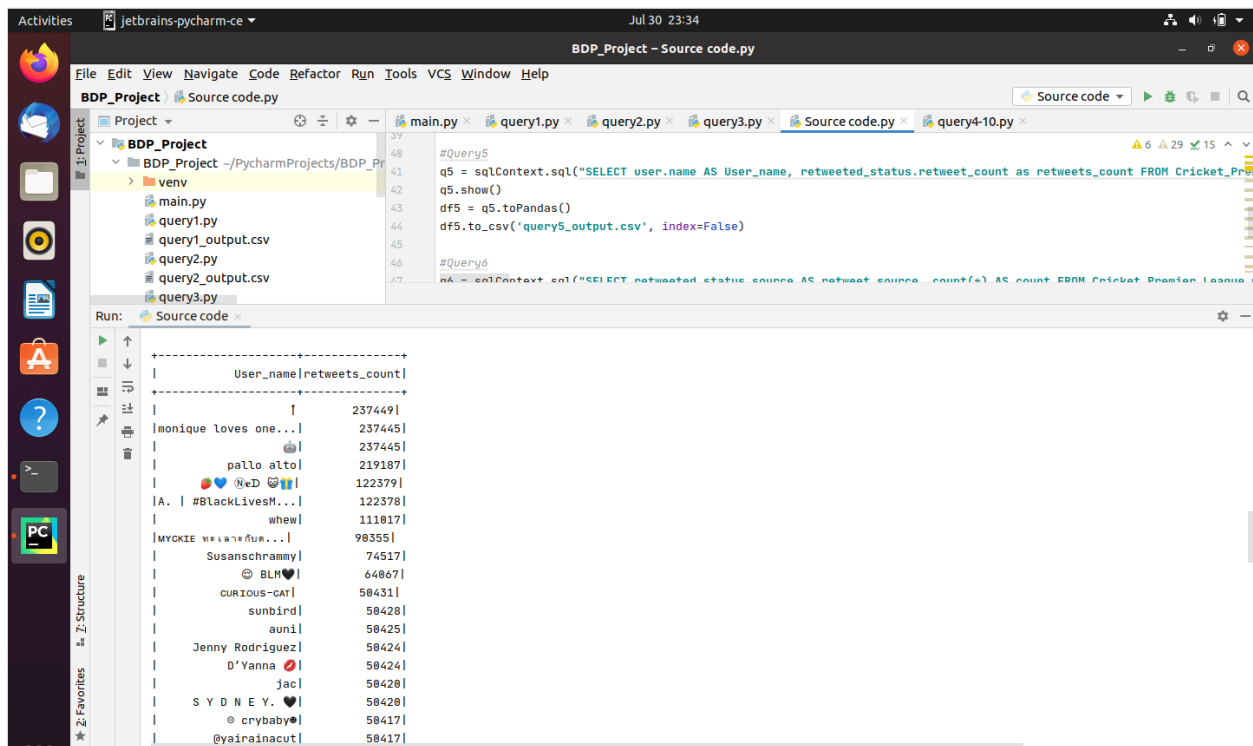
## Query 5:

This query is used to extract the list of users who had retweeted for the tweets and their respective number of retweets.

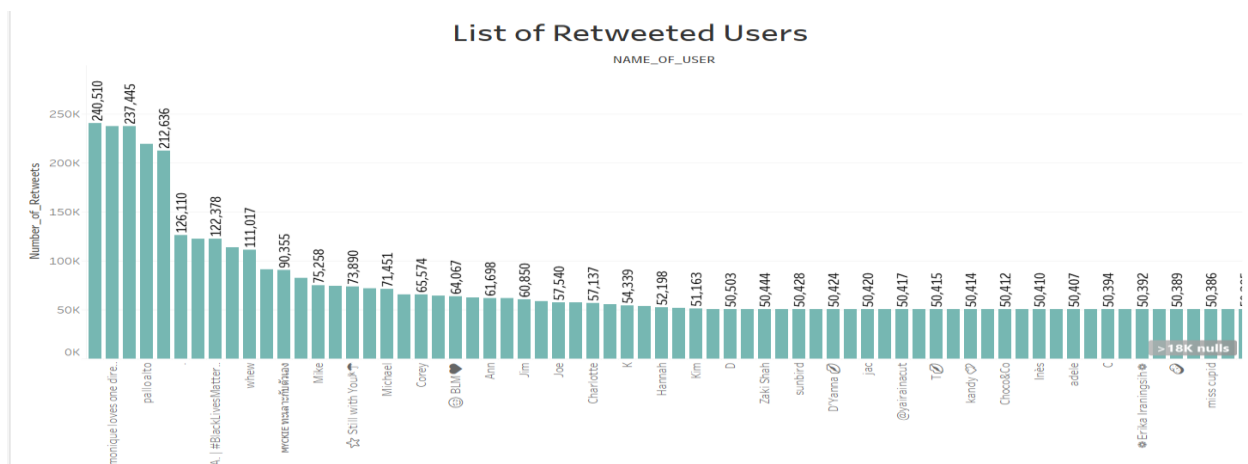
Please find the query below that had been used for extraction

```
q5 = sqlContext.sql("SELECT user.name AS User_name,  
retweeted_status.retweet_count as retweets_count FROM  
Cricket_Premier_League ORDER BY retweets_count DESC")
```

Please find the screenshot of the query output.



Please find the screenshot of the data visualization for query 5.





## Query 6:

This query is used to extract the list of sources from which the tweets had been collected and their count and sort them from highest to lowest based on their count.

Please find the query below that had been used for extraction

```
q6 = sqlContext.sql("SELECT retweeted_status.source AS retweet_source,
count(*) AS count FROM Cricket_Premier_League where
retweeted_status.source IS NOT null GROUP BY retweeted_status.source ORDER
BY count DESC")
```

Please find the screenshot of the query output.

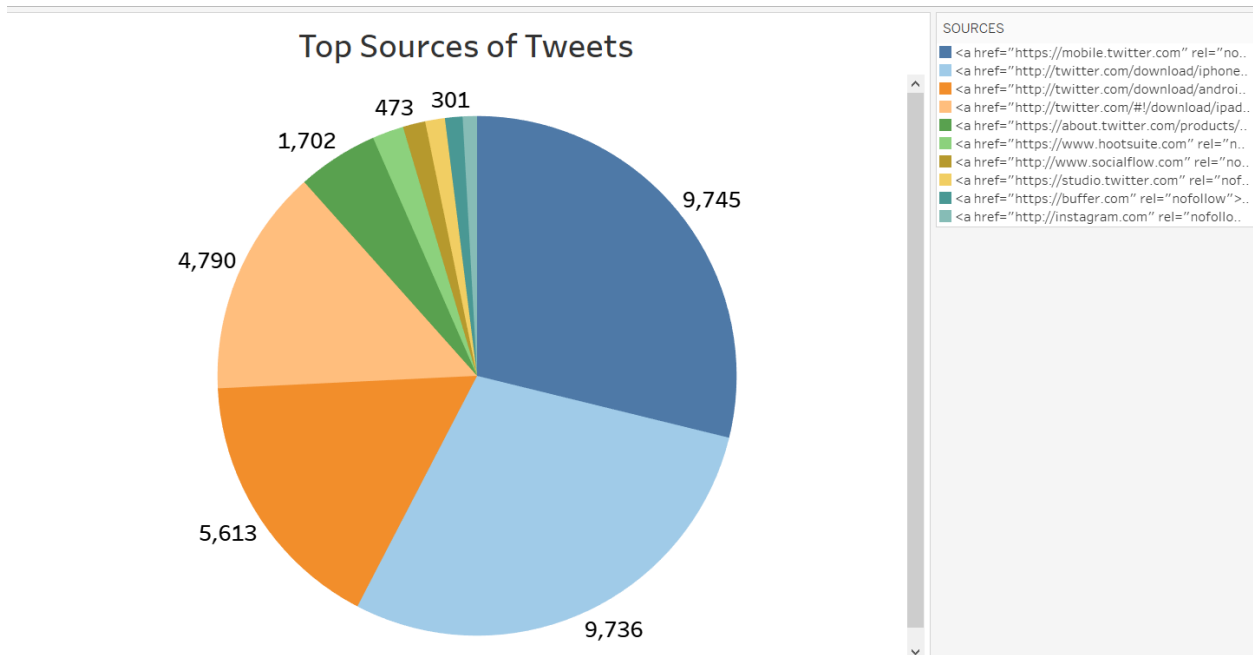
The screenshot shows the PyCharm IDE interface. The top pane displays the source code for a Python script named `query6.py`. The script defines a SQL query `q6` that selects the source of retweeted tweets and counts them, ordered by count in descending order. The bottom pane shows the output of the query, which is a table with two columns: `retweet_source` and `count`. The output lists 20 sources with their corresponding counts, sorted from highest to lowest.

```
#Query6
q6 = sqlContext.sql("SELECT retweeted_status.source AS retweet_source, count(*) AS count FROM Cricket_Premier_League
q6.show()
df6 = q6.toPandas()
df6.to_csv('query6_output.csv', index=False)

#Query7
q7 = sqlContext.sql("SELECT user_id, count(*) AS count FROM Cricket_Premier_League where user_id IS NOT null GROUP BY
```

retweet_source	count
<a href="https://t.me/...">https://t.me/...</a>	9745
<a href="http://t.me/...">http://t.me/...</a>	9736
<a href="http://t.me/...">http://t.me/...</a>	5613
<a href="http://t.me/...">http://t.me/...</a>	4798
<a href="https://t.me/...">https://t.me/...</a>	1782
<a href="https://t.me/...">https://t.me/...</a>	661
<a href="http://w...">http://w...</a>	473
<a href="https://t.me/...">https://t.me/...</a>	415
<a href="https://t.me/...">https://t.me/...</a>	369
<a href="http://t.me/...">http://t.me/...</a>	381
<a href="https://t.me/...">https://t.me/...</a>	195
<a href="https://t.me/...">https://t.me/...</a>	159
<a href="http://w...">http://w...</a>	151
<a href="http://w...">http://w...</a>	115
<a href="https://t.me/...">https://t.me/...</a>	113
<a href="http://w...">http://w...</a>	65
<a href="https://t.me/...">https://t.me/...</a>	55
<a href="https://t.me/...">https://t.me/...</a>	58
<a href="https://t.me/...">https://t.me/...</a>	49
<a href="http://t.me/...">http://t.me/...</a>	48

Please find the screenshot of the data visualization for query 6.



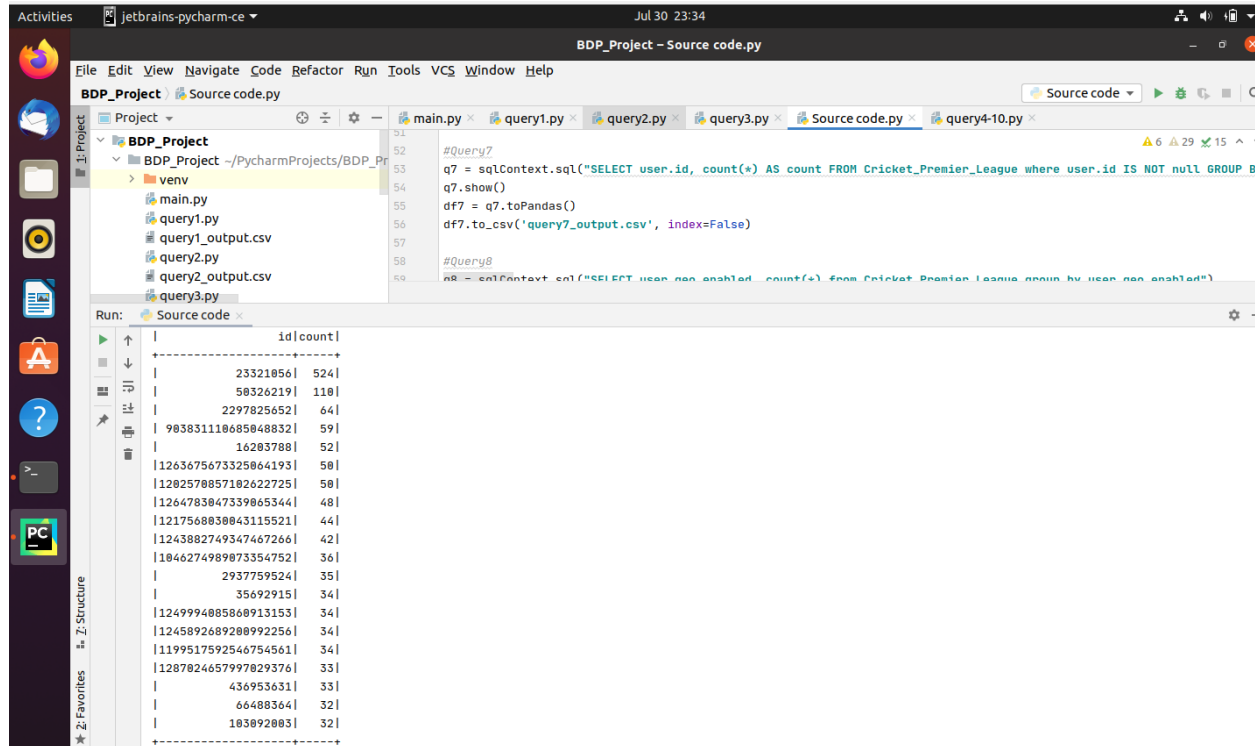
### Query 7:

This query is used to find the user who tweeted the most about cricket.

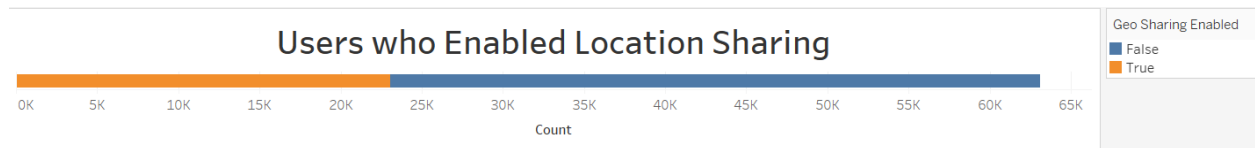
Please find the query below that had been used for extraction

```
q7 = sqlContext.sql("SELECT user.id, count(*) AS count FROM  
Cricket_Premier_League where user.id IS NOT null GROUP BY user.id ORDER BY  
count DESC")
```

Please find the screenshot of the query output.



Please find the screenshot of the data visualization for query 7.




### Query 8:

This query is used to extract the number of users who have enabled and not enabled location sharing.

Please find the query below that had been used for extraction

```
q8 = sqlContext.sql("SELECT user.geo_enabled, count(*) from Cricket_Premier_League group by user.geo_enabled")
```

Please find the screenshot of the query output.



The screenshot shows a PyCharm IDE with a project named 'BDP\_Project'. The file explorer on the left shows a 'venv' directory containing 'main.py', 'query1.py', 'query1\_output.csv', 'query2.py', 'query2\_output.csv', and 'query3.py'. The main editor window shows a Python script with the following code:

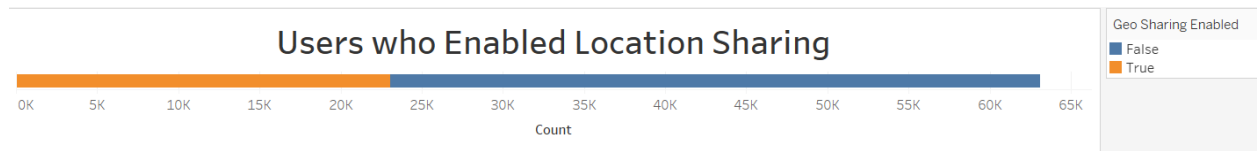
```
#Query8
q8 = sqlContext.sql("SELECT user.geo_enabled, count(*) from Cricket_Premier_League group by user.geo_enabled")
q8.show()
df8 = q8.toPandas()
df8.to_csv('query8_output.csv', index=False)

#Query9
q9 = sqlContext.sql("SELECT possibly_sensitive, count(*) AS count FROM Cricket_Premier_League GROUP BY possibly_sensitive")
```

The 'Run' console at the bottom shows the output of the query:

```
+-----+-----+
|geo_enabled|count(1)|
+-----+-----+
|      true|   23065|
|     false|   40022|
+-----+-----+
```

Please find the screenshot of the data visualization for query 8.



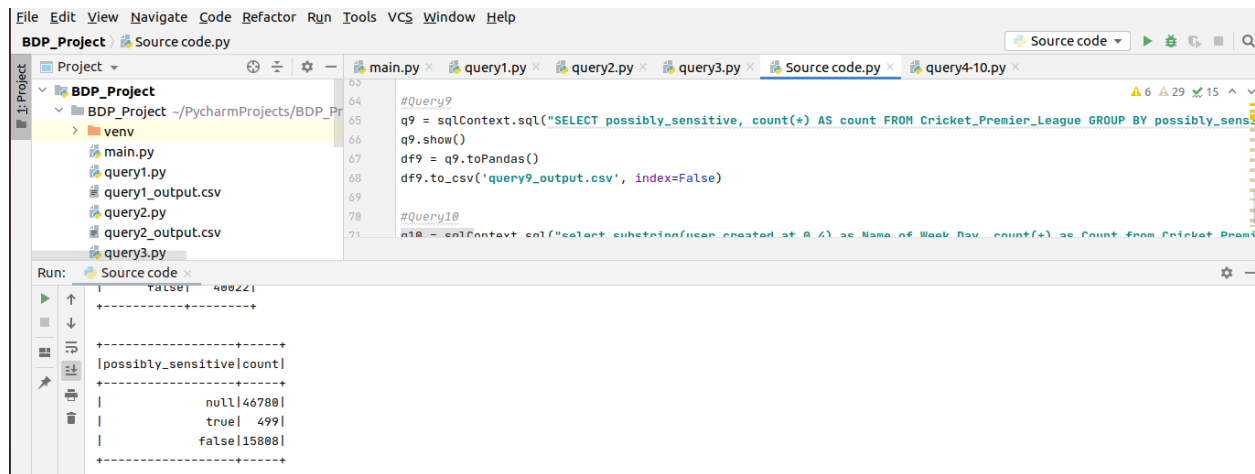
## Query 9:

This query is used to test the sensitive data.

Please find the query below that had been used for extraction

```
q9 = sqlContext.sql("SELECT possibly_sensitive, count(*) AS count FROM Cricket_Premier_League GROUP BY possibly_sensitive")
```

Please find the screenshot of the query output.

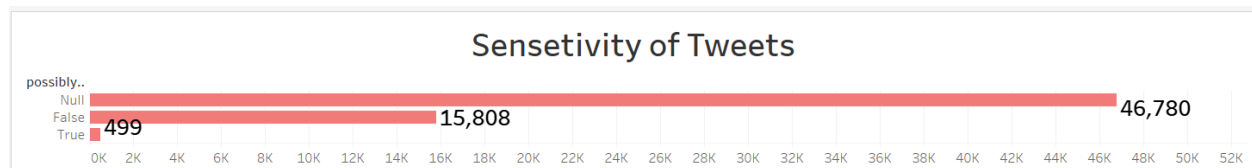


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
BDP_Project Source code.py
Project
BDP_Project
BDP_Project --PycharmProjects/BDP_Pr
venv
main.py
query1.py
query1_output.csv
query2.py
query2_output.csv
query3.py
main.py
query1.py
query2.py
query3.py
Source code.py
query4-10.py
#Query9
q9 = sqlContext.sql("SELECT possibly_sensitive, count(*) AS count FROM Cricket_Premier_League GROUP BY possibly_sens
q9.show()
df9 = q9.toPandas()
df9.to_csv('query9_output.csv', index=False)
#Query10
q10 = sqlContext.sql("select substring(user.created_at,0,4) as Name_of_Week_Day, count(*) as Count from Cricket_Premie
q10.show()
df10 = q10.toPandas()
df10.to_csv('query10_output.csv', index=False)
```

Run: Source code

possibly_sensitive	count
null	46788
true	499
false	15808

Please find the screenshot of the data visualization for query 9.



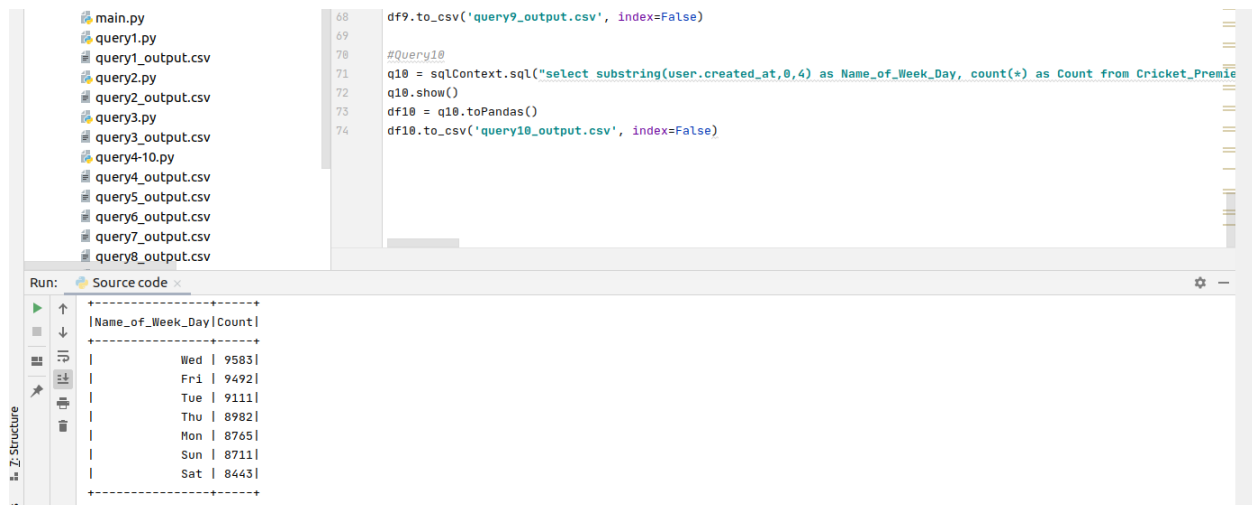
## Query 10:

This query is used to find the weekdays on which tweet activity is high.

Please find the query below that had been used for extraction

```
q10 = sqlContext.sql("select substring(user.created_at,0,4) as
Name_of_Week_Day, count(*) as Count from Cricket_Premier_League group by
substring(user.created_at,0,4) order by Count desc")
```

Please find the screenshot of the query output.



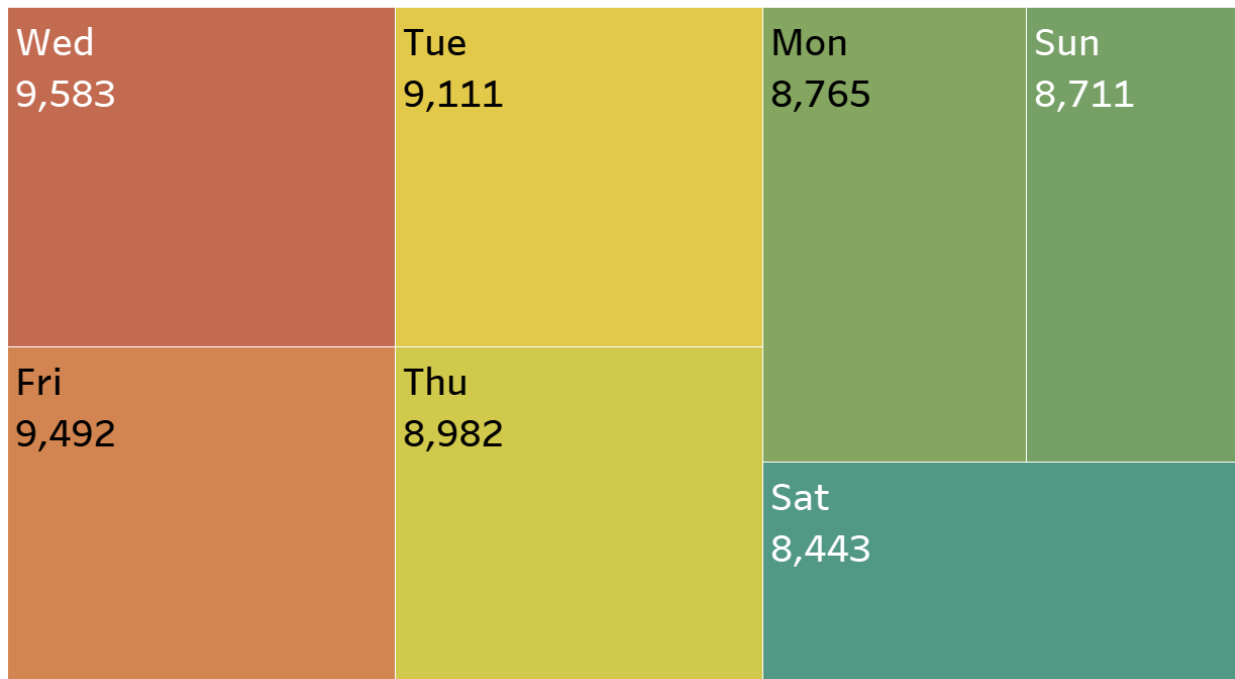
```
main.py
query1.py
query1_output.csv
query2.py
query2_output.csv
query3.py
query3_output.csv
query4-10.py
query4_output.csv
query5_output.csv
query6_output.csv
query7_output.csv
query8_output.csv
df9.to_csv('query9_output.csv', index=False)
#Query10
q10 = sqlContext.sql("select substring(user.created_at,0,4) as Name_of_Week_Day, count(*) as Count from Cricket_Premie
q10.show()
df10 = q10.toPandas()
df10.to_csv('query10_output.csv', index=False)
```

Run: Source code

Name_of_Week_Day	Count
Wed	9583
Fri	9492
Tue	9111
Thu	8982
Mon	8765
Sun	8711
Sat	8443

Please find the screenshot of the data visualization for query 10.

### Activity on Weekdays



Please find the Github link below

<https://github.com/ynkc3/Principles-of-Big-Data-Academic-Project/tree/master/Phase%202>