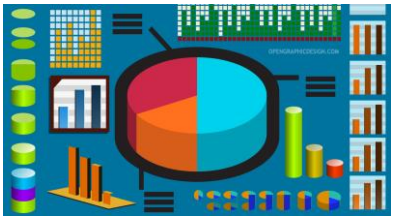
PB(11/14/2017)

Principles of Big Data Management
Phase-2

Fall 2017



Submitted by: (Team 13)

Pragathi Thammaneni
Sridevi Mallipudi
Fatema Hasta

Analyzing and Visualizing Twitter data on Mobiles

Objective:

Applications/Software's Used: Apache Spark SQL, Scala, Cloudera, Twitter Developer Account, Python.

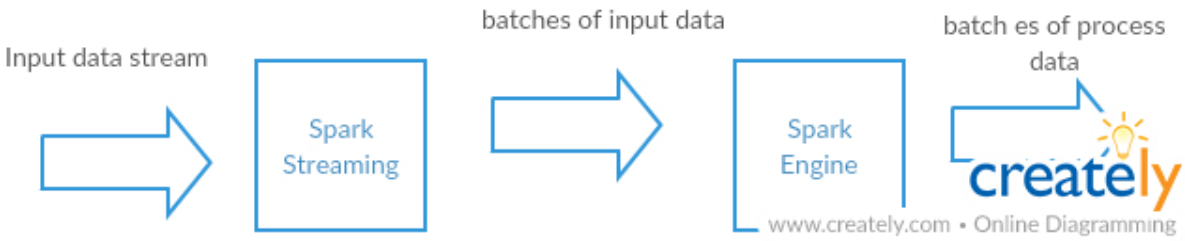
Collecting tweets from Twitter:

- Firstly, we have created a developer account in Twitter using below link. https://apps.twitter.com/
- Below are the variables that contains the user credentials to access Twitter API

  ➢ API ACCESS_TOKEN = " 1329248834-R3X8SrLzchMQyBOSRtV1QSJYfis2lidRex4RXeq"
  ➢ ACCESS_SECRET = " g6pwJFjRJTQK2cRi01rGV3kpr6C9v7sQNTjrtPLtiETc4"
  ➢ CONSUMER_KEY = " l82JLsLffFzGMrytXhF4DAnf2"
  ➢ CONSUMER_SECRET = " Fh4985bvDNeF2RurZsseuDYBnbFv3cLmpdyyizFapBkr1uS954"
- We have written python program that is used to fetch tweets in JSON format. (TweetsExtract.py)
- The tweet data is collected on the concept based on to analyze and visualize the data regarding various mobile phones.

```
Sample tweets are collected for the key words by using python program
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

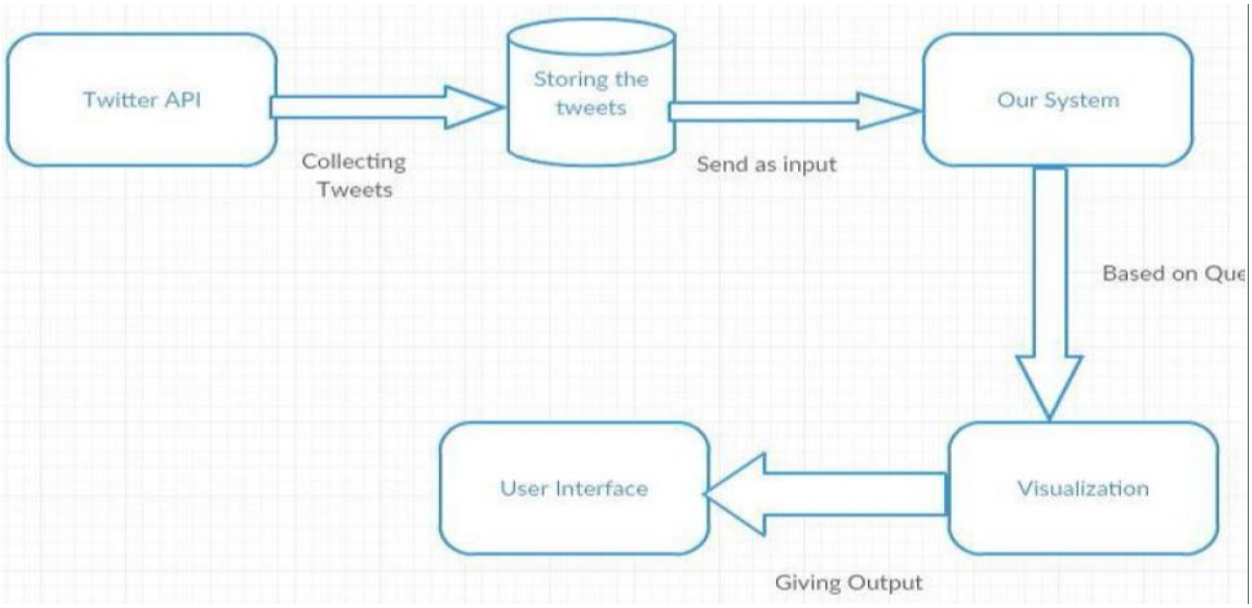
1) Spark streaming ,the data can be -ingested – from different sources like twitter and can perform high level complex algorithms like queries and the processed data can be pushed out to the file systems.To stream the twitter data twitter.utils contains all the built in functionality -to- stream data from twitter.



- **SqlContext which is a new- session, with separated SQL configurations, registered functions temporary tables (data frames to store the all relational functionality in Spark SQL).**

The main theme of this project is to do big data analytics on the Mobile phones. Based on the twitter tweets, we predicted few interesting query analysis and visualization on mobile phone twitter data. First, we collected the tweets regarding the mobile phone data from the twitter API. By using the collected data, we build 10 interesting queries which results a data analysis on the twitter data. Visualization is done for on the result sets and viewed in some pictorial representations like pie chart and bar graphs.

Architecture diagram:



Sample JSON Object Structure:

The tweets are saved and stored with the object format as JSON.

{

"created_at":"Fri Nov 10 16:11:34 +0000 2017",

"id":929018704041349120,

"id_str":"929018704041349120",

"text":"You know #SteveJobs hustled his way up to @Apple (and they still got the best hustle! You got that #iphoneX don\u2019t y\u2026 https:\/\/t.co\/GqUSbaqhFF",

"display_text_range":[

0,

140

],

"source":"\u003ca href=\"https:\/\/mobile.twitter.com\" rel=\"nofollow\"\u003eTwitter Lite\u003c\/a\u003e",

"truncated":true,

"in_reply_to_status_id":null,

"in_reply_to_status_id_str":null,

"in_reply_to_user_id":null,

"in_reply_to_user_id_str":null,

"in_reply_to_screen_name":null,

"user":{

"id":923408384358940674,

"id_str":"923408384358940674",

"name":"Retired Ratchet",

"screen_name":"retiredratchet",

"location":"Las Vegas, NV",

"url":null,

"description":"#Blitter for the culture:\nRatchet is a slang term in hip hop that in the strictest sense refers to an uncouth female & is a Louisianan regiolect of \"wretched\"",

"translator_type":"none",

"protected":false,

"verified":false,

"followers_count":156,

"friends_count":224,

listed_count":0,

"favourites_count":993,

"statuses_count":618,

"created_at":"Thu Oct 26 04:38:09 +0000 2017",

"utc_offset":null,

"time_zone":null,

"geo_enabled":false,

"lang":"en",

"contributors_enabled":false,

"is_translator":false,

"profile_background_color":"F5F8FA",

"profile_background_image_url":"",

"profile_background_image_url_https":"",

"profile_background_tile":false,

"profile_link_color":"1DA1F2",

"profile_sidebar_border_color":"C0DEED",

"profile_sidebar_fill_color":"DDEEF6",

"profile_text_color":"333333",

"profile_use_background_image":true,

"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/923427538038173696\/tBE9KGIV_normal.jpg",

"profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/923427538038173696\/tBE9KGIV_normal.jpg",

"profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/923408384358940674\/1510329291",

"default_profile":true,

"default_profile_image":false,

"following":null,

"follow_request_sent":null,

"notifications":null

},

"geo":null,

"coordinates":null,

"place":null,

"contributors":null,

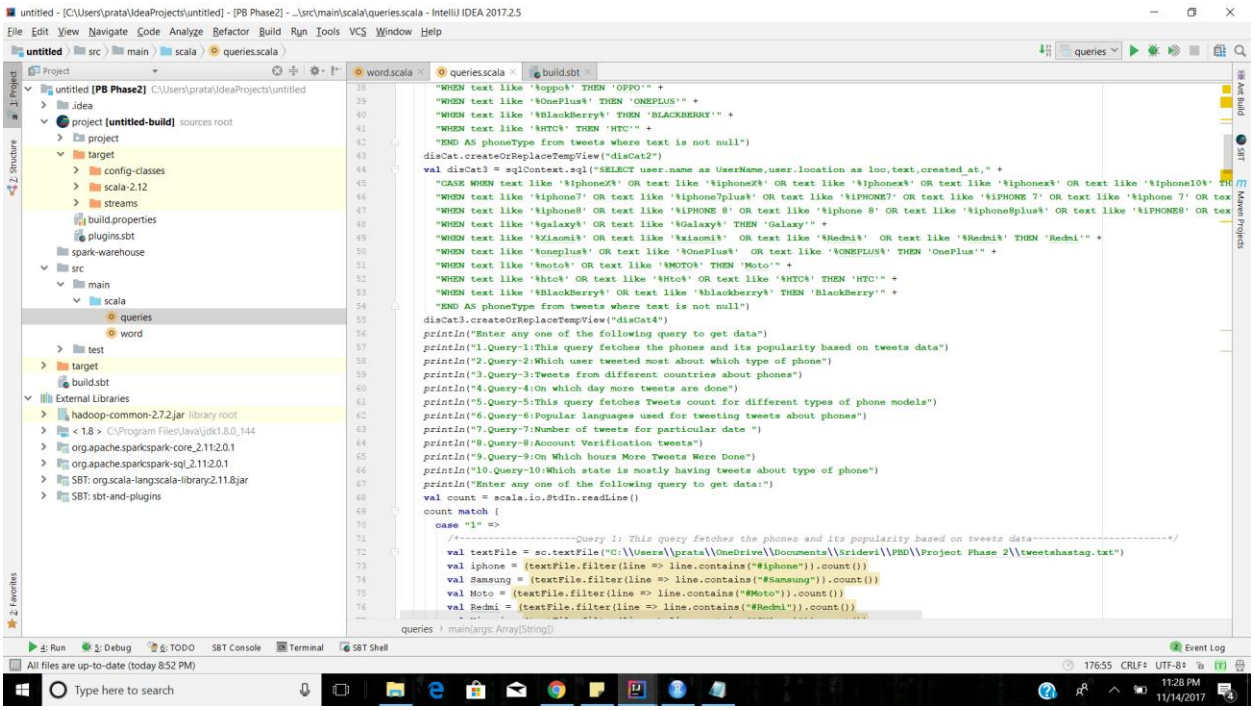
"quoted_status_id":927672133739835392,..

}

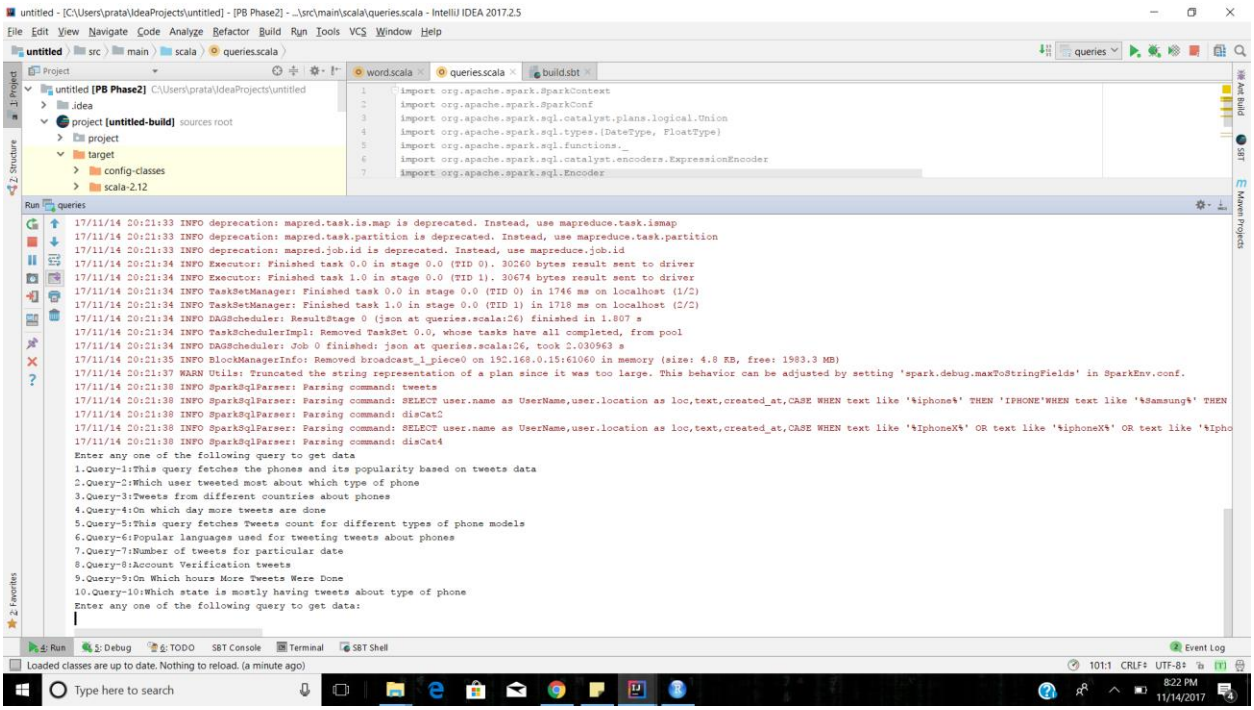
2)Queries and outputs:

Below console shows indexes for queries to perform which query by the input choice given, to get the corresponding data.

Code:



Output:



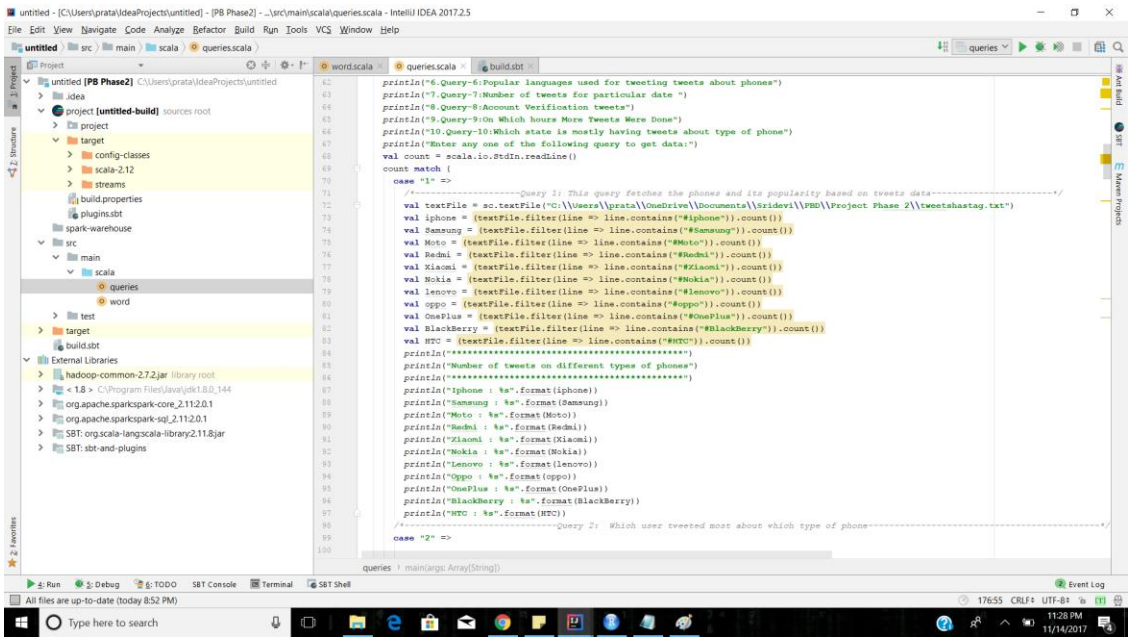
Query 1: Query for fetching the tweets corresponding to phones and its popularity count on the tweets depending up the on tweets data collected.

Answer: This query is built to analyze the tweets – what the count of each model which are tweeted by the users in the collected tweets which directly reflects the popularity of each phone model.

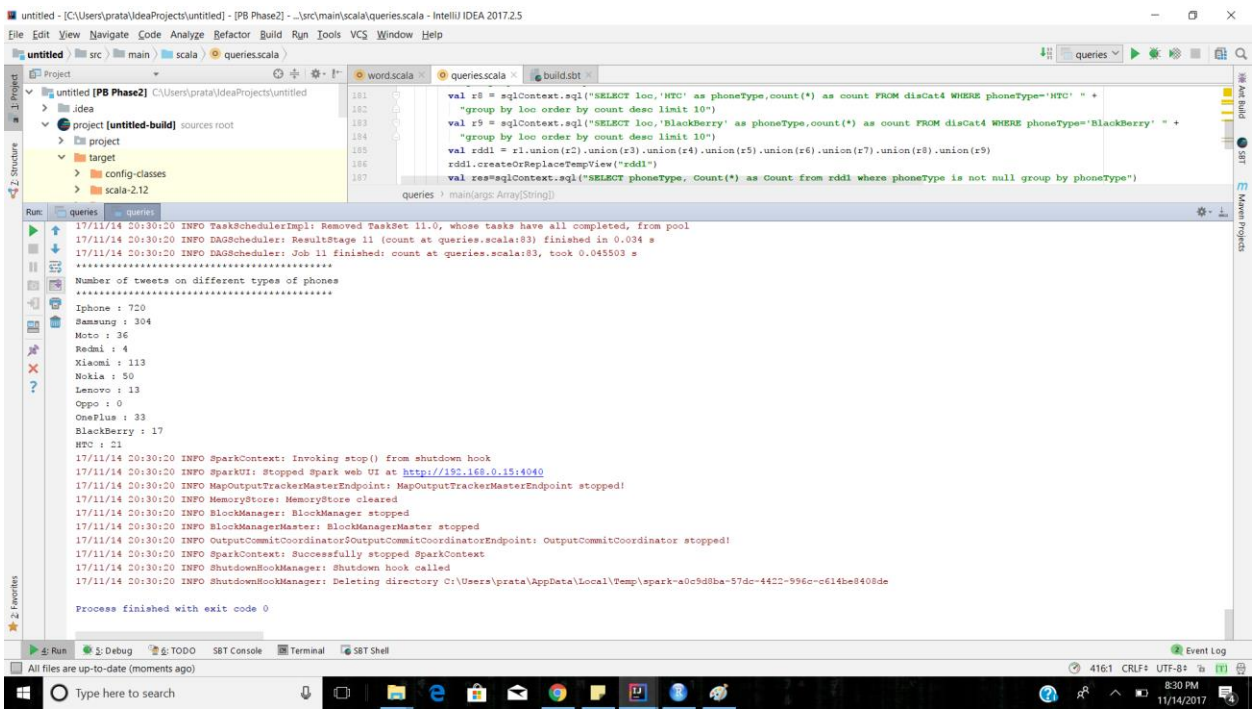
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to perform the count operation on each model. So, it results the count -how many times a phone model appears in the tweets.

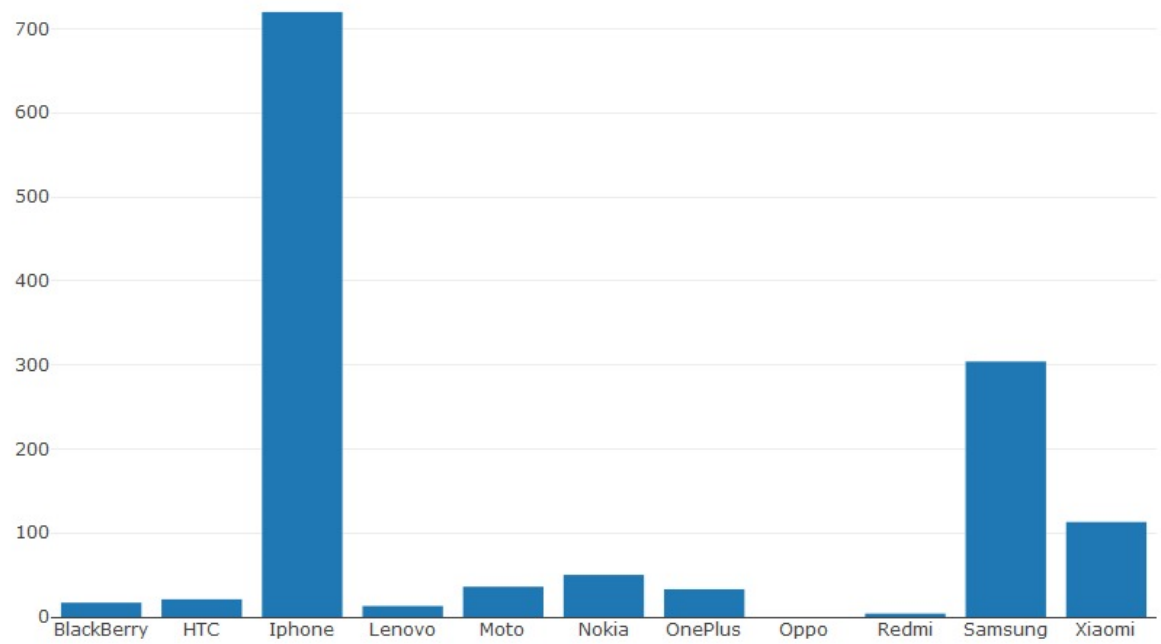
Code:


println("6.Query-6:Popular languages used for tweeting tweets about phones")
println("7.Query-7:Number of tweets for particular date ")
println("8.Query-8:Account Verification tweets")
println("9.Query-9:On Which hours More Tweets Were Done")
println("10.Query-10:Which state is mostly having tweets about type of phone")
println("Enter any one of the following query to get data:")
val count = scala.io.StdIn.readLine()
count match {
case "1" =>
/*----------Query 1: This query fetches the phones and its popularity based on tweets data----------*/
val textFile = sc.textFile("C:\\Users\\prata\\OneDrive\\Documents\\Sridevi\\PBD\\Project Phase 2\\tweetshastag.txt")
val iphone = (textFile.filter(line => line.contains("#iphone")).count())
val Samsung = (textFile.filter(line => line.contains("#Samsung")).count())
val Moto = (textFile.filter(line => line.contains("#Moto")).count())
val Redmi = (textFile.filter(line => line.contains("#Redmi")).count())
val Xiaomi = (textFile.filter(line => line.contains("#Xiaomi")).count())
val Nokia = (textFile.filter(line => line.contains("#Nokia")).count())
val lenovo = (textFile.filter(line => line.contains("#lenovo")).count())
val oppo = (textFile.filter(line => line.contains("#oppo")).count())
val OnePlus = (textFile.filter(line => line.contains("#OnePlus")).count())
val BlackBerry = (textFile.filter(line => line.contains("#BlackBerry")).count())
val HTC = (textFile.filter(line => line.contains("#HTC")).count())
println("**********")
println("Number of tweets on different types of phones")
println("**********")
println("Iphone : %s".format(iphone))
println("Samsung : %s".format(Samsung))
println("Moto : %s".format(Moto))
println("Redmi : %s".format(Redmi))
println("Xiaomi : %s".format(Xiaomi))
println("Nokia : %s".format(Nokia))
println("Lenovo : %s".format(lenovo))
println("Oppo : %s".format(oppo))
println("OnePlus : %s".format(OnePlus))
println("BlackBerry : %s".format(BlackBerry))
println("HTC : %s".format(HTC))
/*----------Query 2: Which user tweeted most about which type of phone----------*/
case "2" =>
queries › main(args: Array[String])

Output:


val r8 = sqlContext.sql("SELECT loc,'HTC' as phoneType,count(*) as count FROM disCat4 WHERE phoneType='HTC' " +
"group by loc order by count desc limit 10")
val r9 = sqlContext.sql("SELECT loc,'BlackBerry' as phoneType,count(*) as count FROM disCat4 WHERE phoneType='BlackBerry' " +
"group by loc order by count desc limit 10")
val rdd1 = r1.union(r2).union(r3).union(r4).union(r5).union(r6).union(r7).union(r8).union(r9)
rdd1.createOrReplaceTempView("rdd1")
val res=sqlContext.sql("SELECT phoneType, Count(*) as Count from rdd1 where phoneType is not null group by phoneType")
queries › main(args: Array[String])
17/11/14 20:30:20 INFO TaskSchedulerImpl: Removed TaskSet 11.0, whose tasks have all completed, from pool
17/11/14 20:30:20 INFO DAGScheduler: ResultStage 11 (count at queries.scala:83) finished in 0.034 s
17/11/14 20:30:20 INFO DAGScheduler: Job 11 finished: count at queries.scala:83, took 0.045503 s
****
Number of tweets on different types of phones
****
Iphone : 720
Samsung : 304
Moto : 36
Redmi : 4
Xiaomi : 113
Nokia : 50
Lenovo : 13
Oppo : 0
OnePlus : 33
BlackBerry : 17
HTC : 21
17/11/14 20:30:20 INFO SparkContext: Invoking stop() from shutdown hook
17/11/14 20:30:20 INFO SparkUI: Stopped Spark web UI at http://192.168.0.15:4040
17/11/14 20:30:20 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
17/11/14 20:30:20 INFO MemoryStore: MemoryStore cleared
17/11/14 20:30:20 INFO BlockManager: BlockManager stopped
17/11/14 20:30:20 INFO BlockManagerMaster: BlockManagerMaster stopped
17/11/14 20:30:20 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
17/11/14 20:30:20 INFO SparkContext: Successfully stopped SparkContext
17/11/14 20:30:20 INFO ShutdownHookManager: Shutdown hook called
17/11/14 20:30:20 INFO ShutdownHookManager: Deleting directory C:\Users\prata\AppData\Local\Temp\spark-a0c9d8ba-57dc-4422-996c-c614be8408de
Process finished with exit code 0

Visualization:



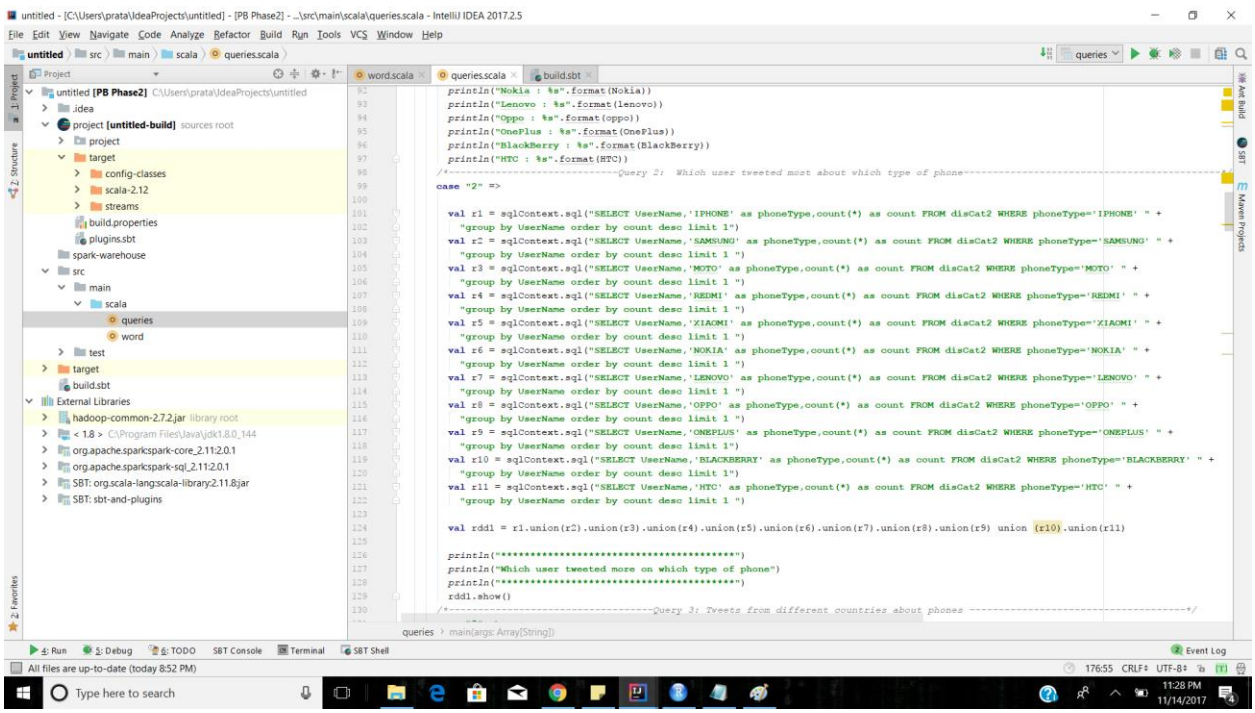
Query 2: Query for fetching which user tweeted most about which type of phone.

Answer: This query is built to analyze the tweets – based on which user tweeted most about which type of phone.

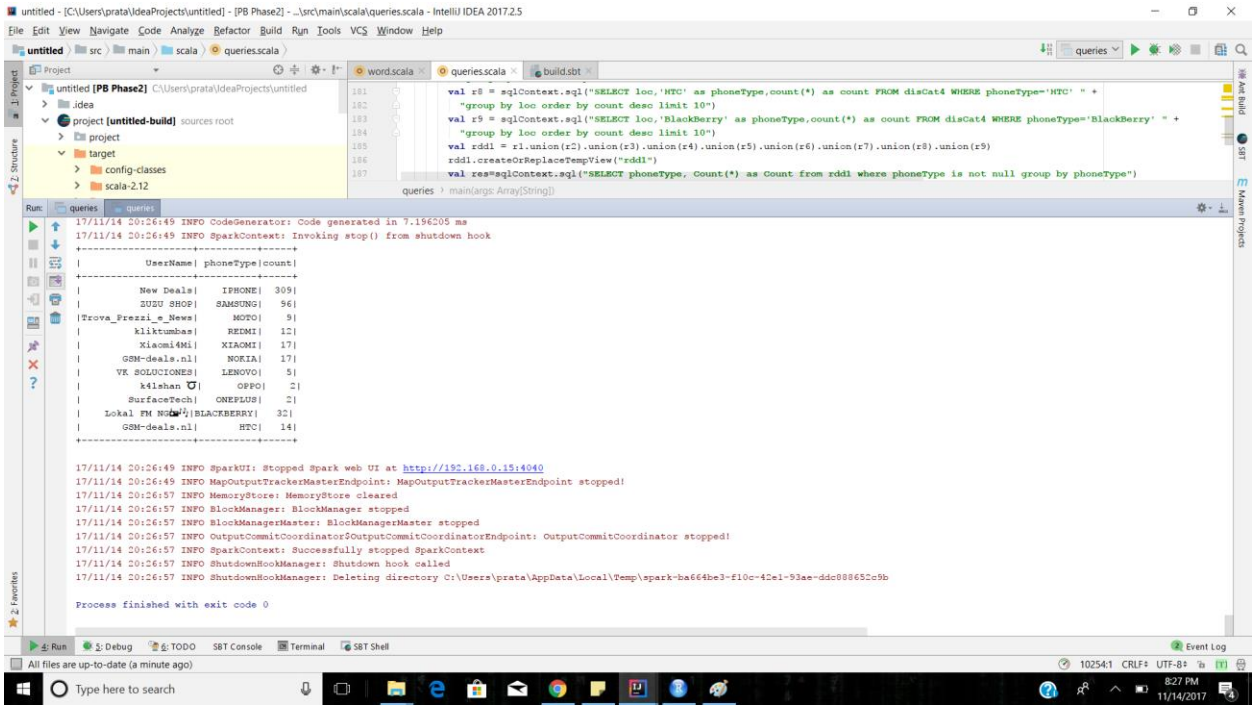
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to perform the user tweeted most about which type of phone. So, it results the count -how many times a user tweeted at most for each mobile phone.

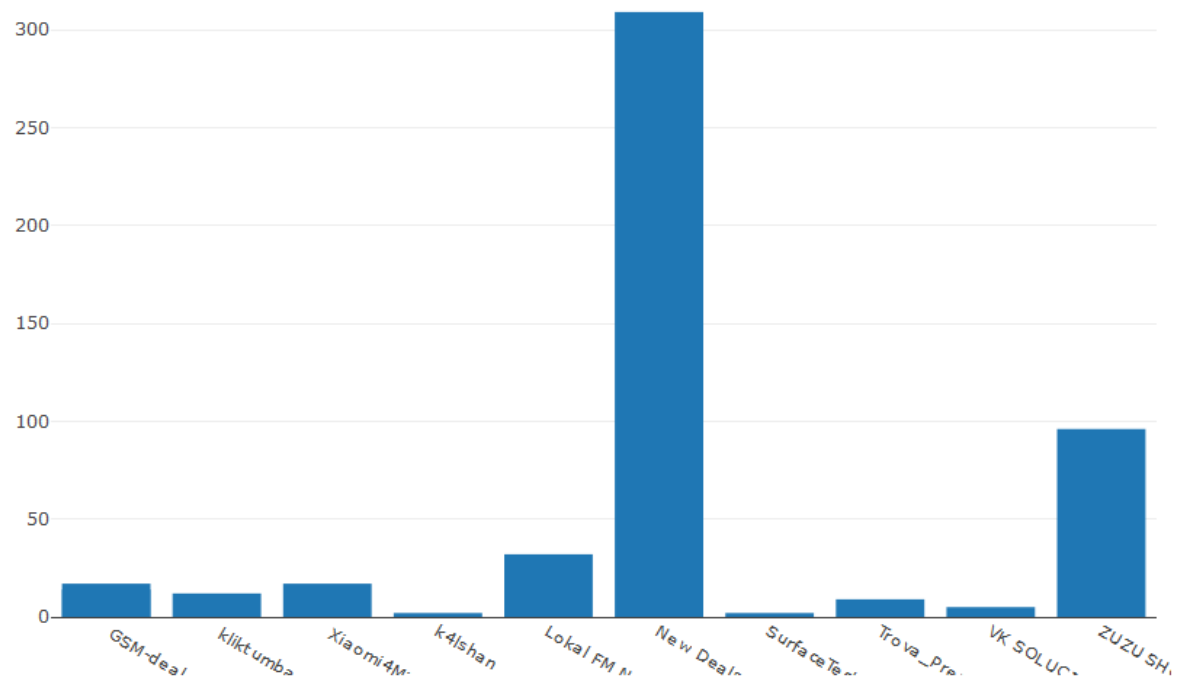
Code:



Output:



Visualization:



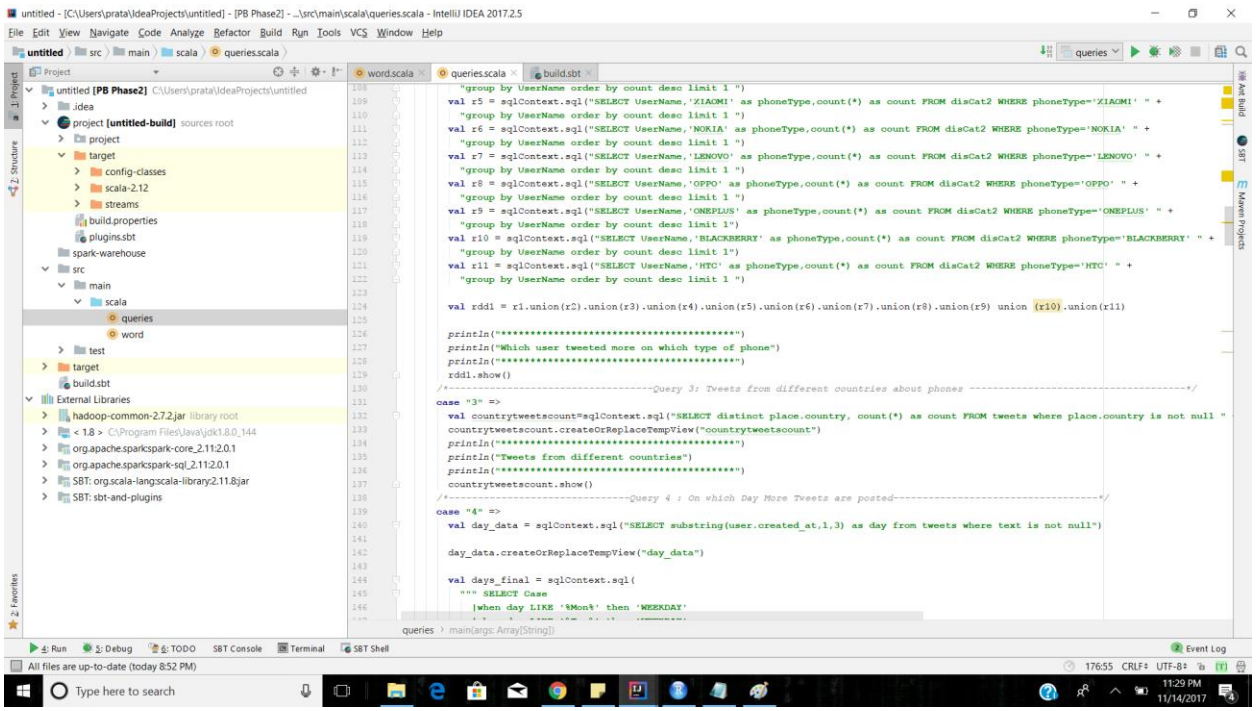
Query 3: Query for fetching Tweets from different countries about phones

Answer: This query is built to analyze the tweets – based on which country tweeted more about on phones.

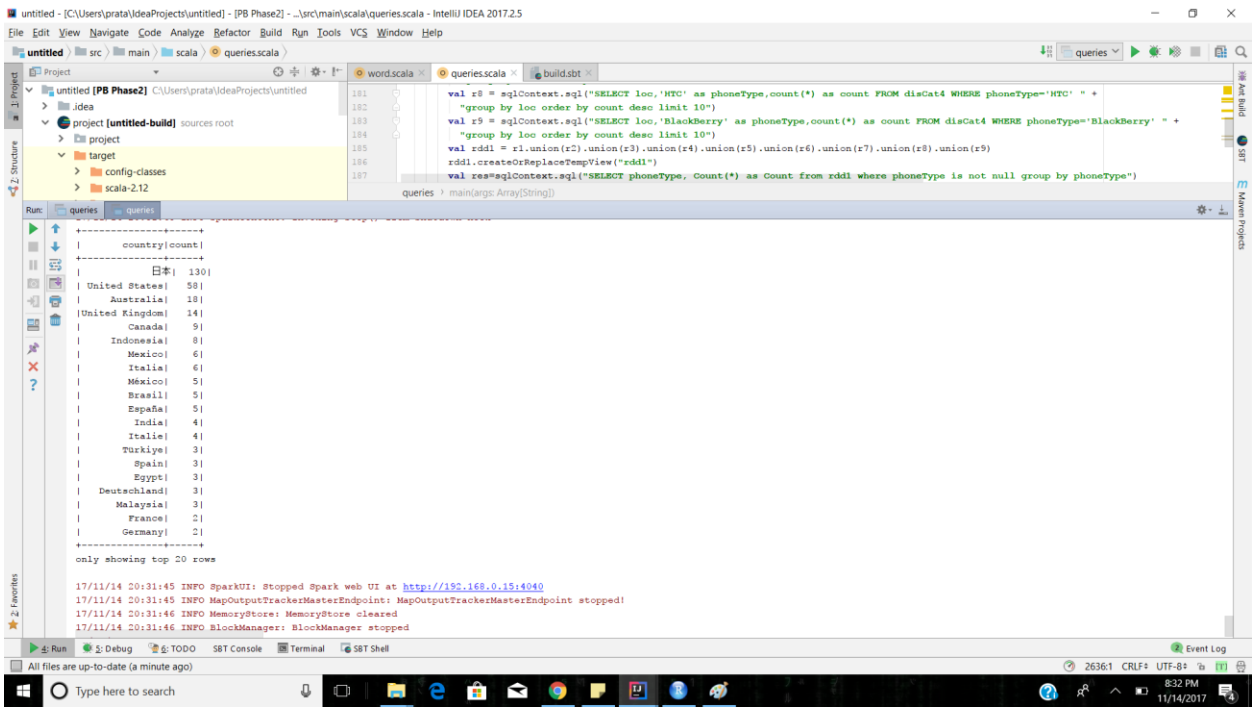
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query for finding out tweets from different countries about phones. So, it results the count -how many tweets from different countries posted about phones.

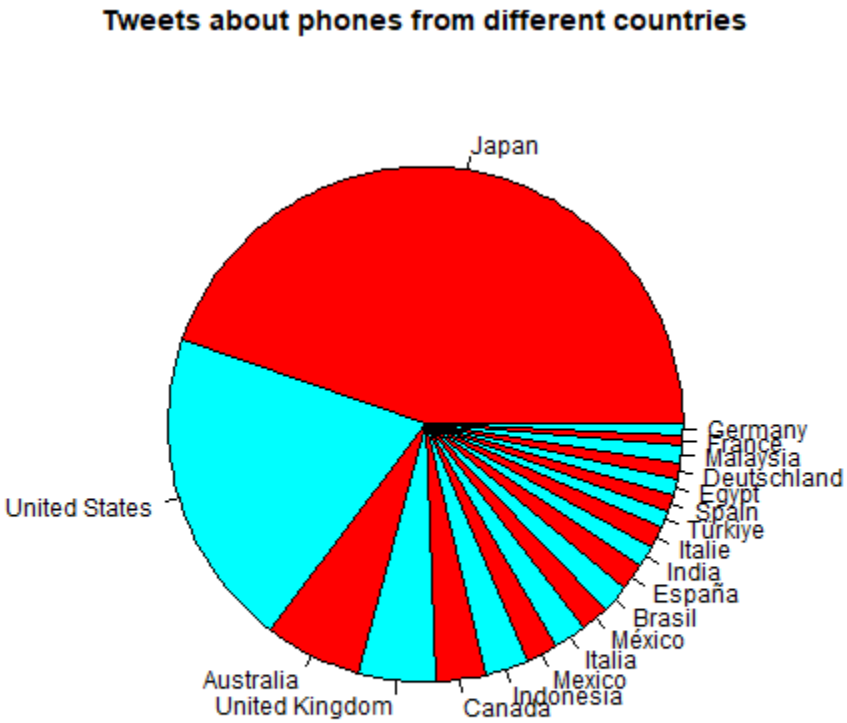
Code:



Output:



Visualization:



Query 4: Query for on which day more tweets are done.

Answer: This query is built to analyze the tweets – based on which day more tweets are done.

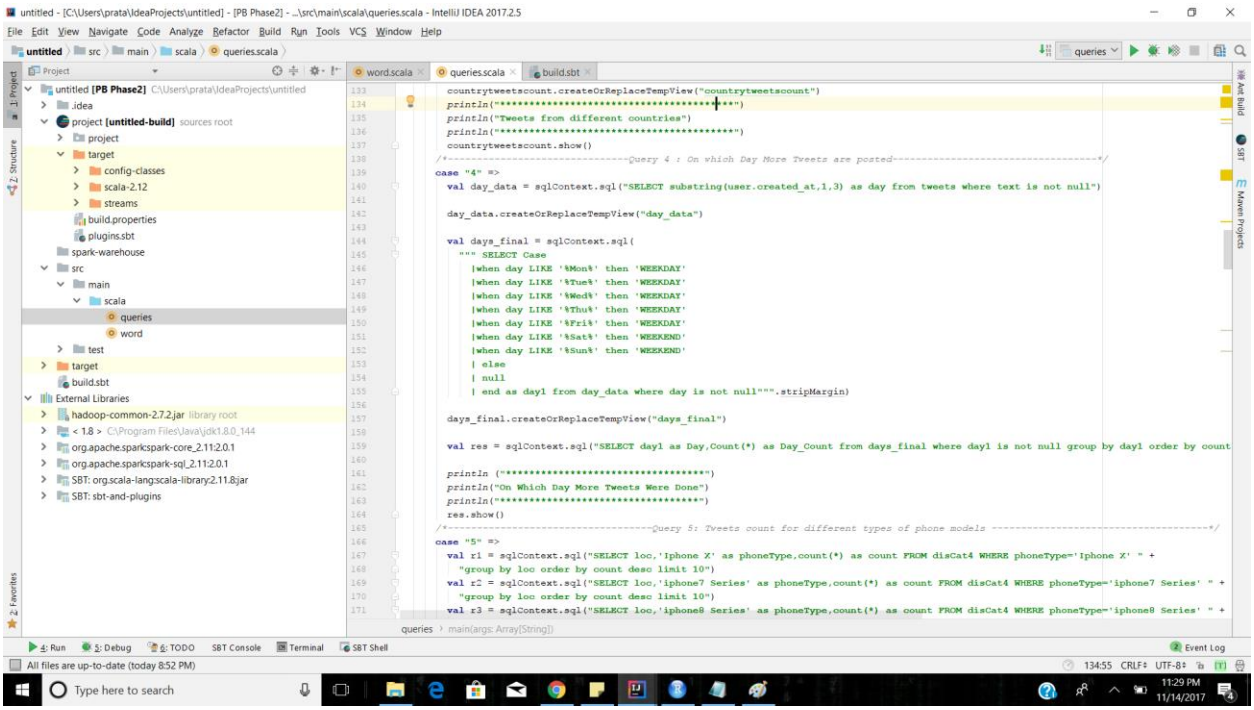
MONDAY, TUESDAY, WEDNESDAY, THURSDAY,FRIDAY as Weekday

SATURDAY, SUNDAY as Weekend

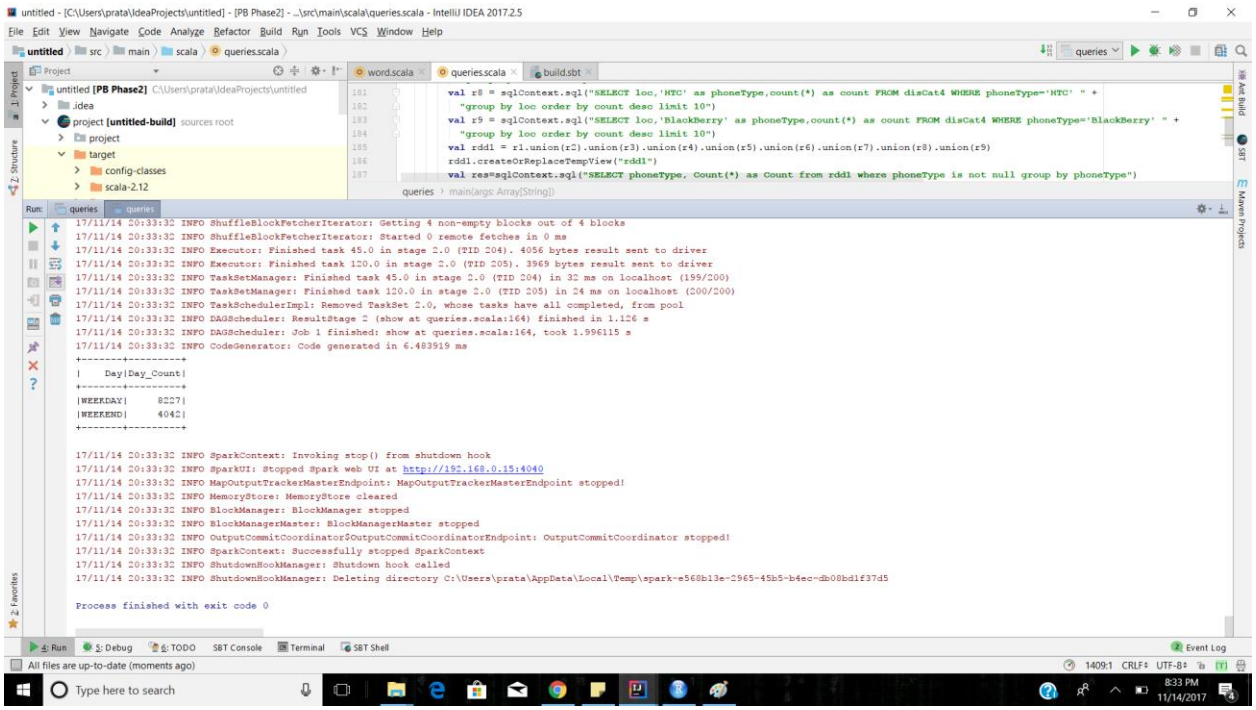
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to perform for analyzing on which day the more tweets are posted. So, it results the count – about giving a figure of on what day & how many tweets are done.

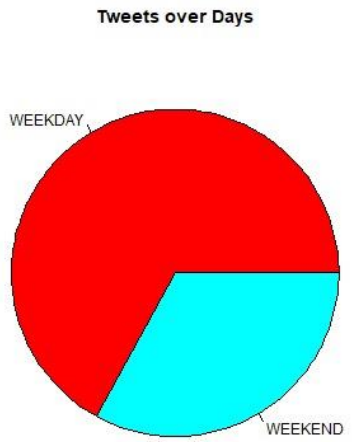
Code:



Output:



Visualization:



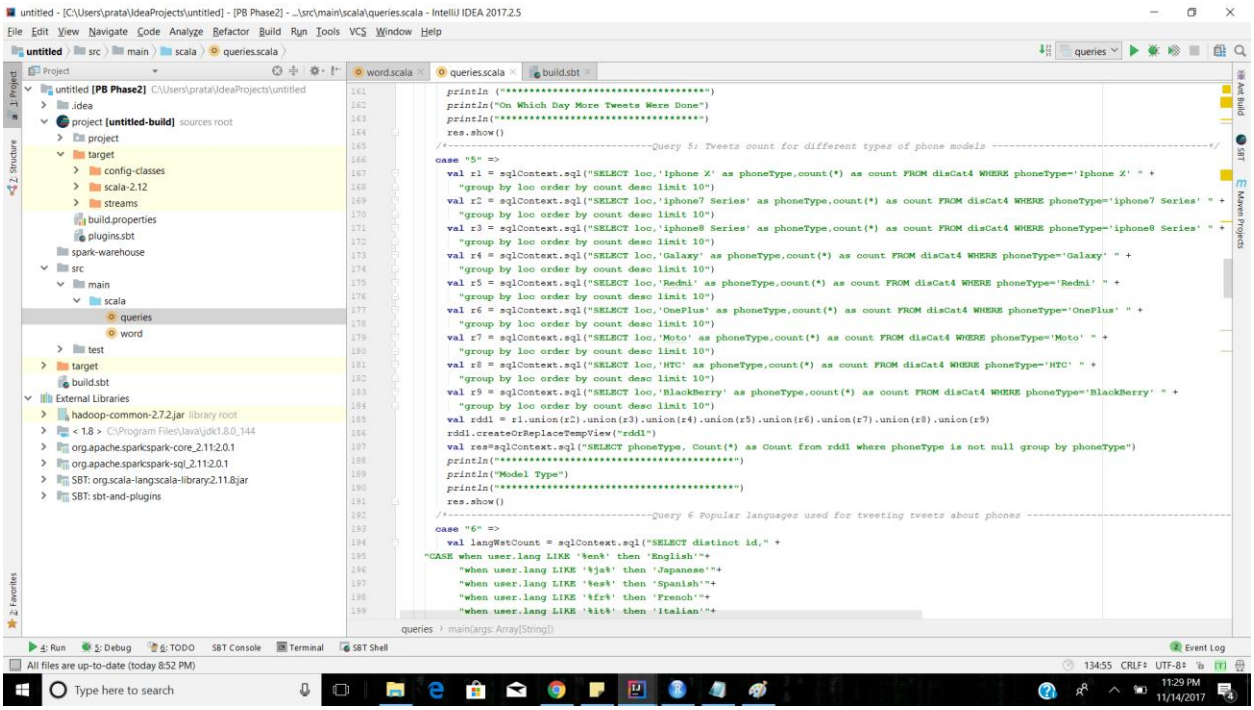
Query 5: This query fetches Tweets count for different series of phone models.

Answer: This query is built to analyze the tweets – for the individual series of each mobile phone.

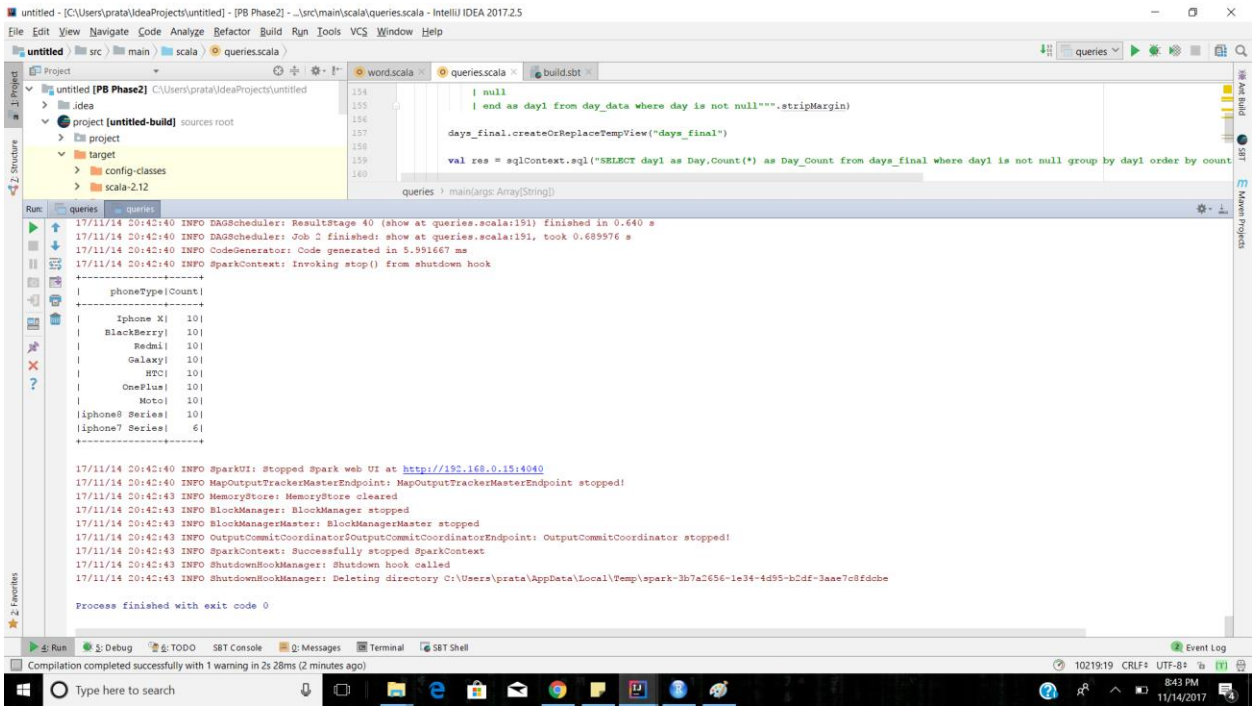
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to perform the user tweeted most which type of series for mobile phone. So, it results the tweets count -for different series of phone models.

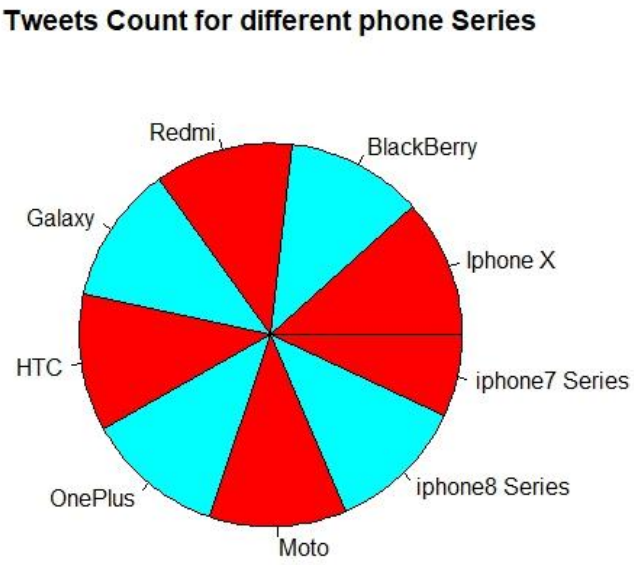
Code:



Output:



Visualization:



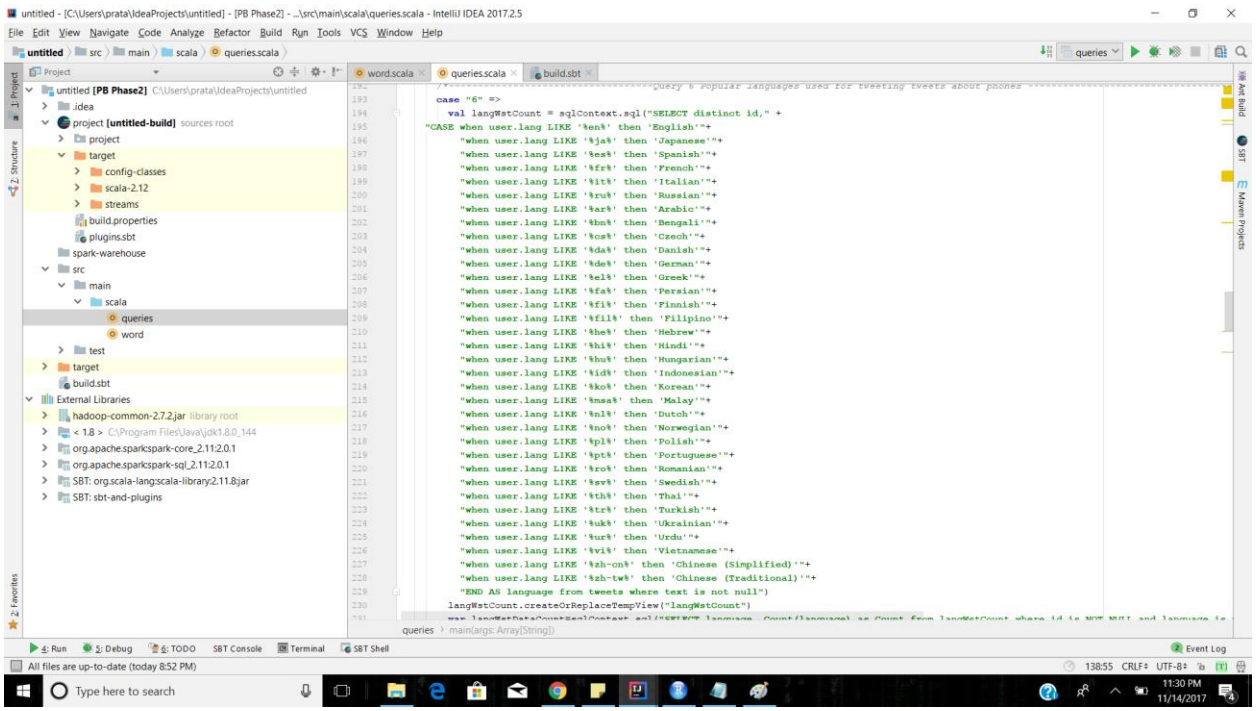
Query 6: Query for fetching Popular languages used for tweeting tweets about phones.

Answer: This query is built to analyze the tweets – popular languages used for tweeting tweets about phones.

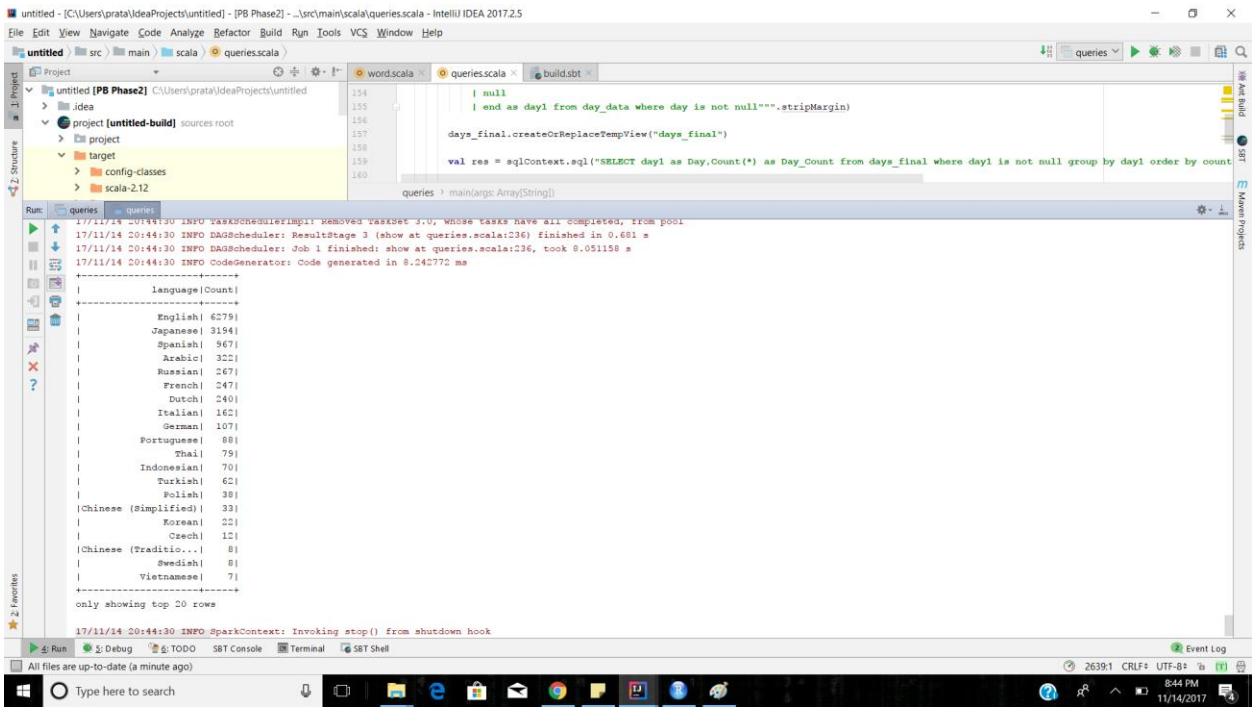
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to analyze the popular languages used. So, it results the count – of most used language by the users.

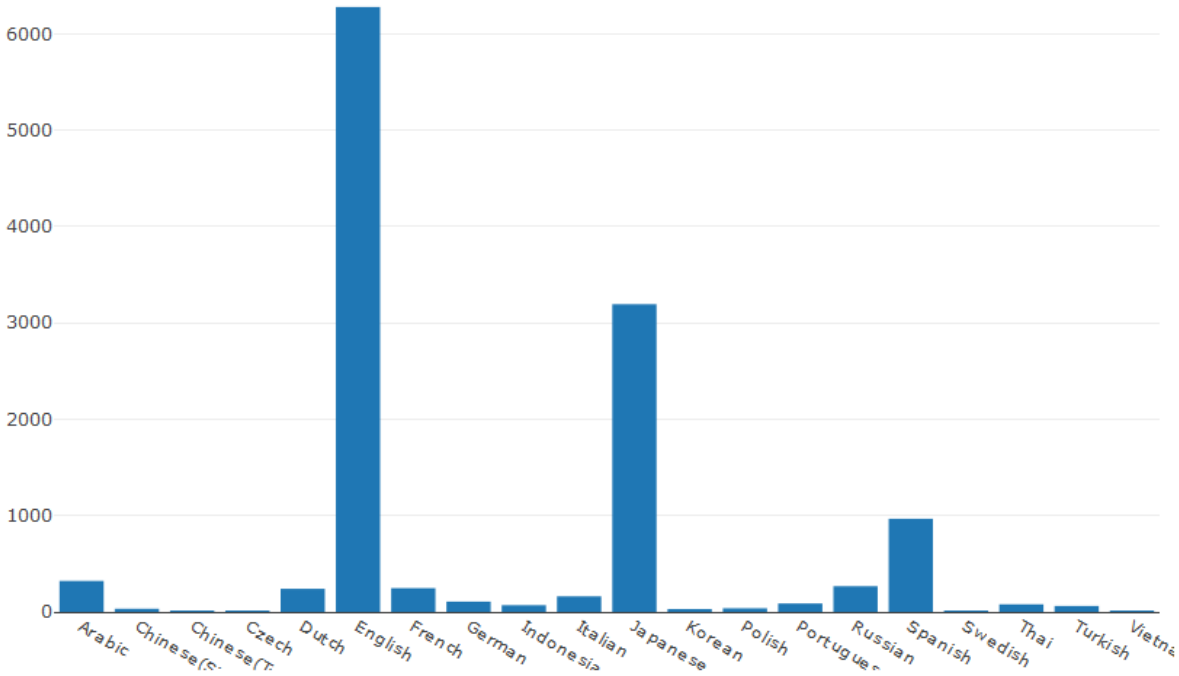
Code:



Output:



Visualization:



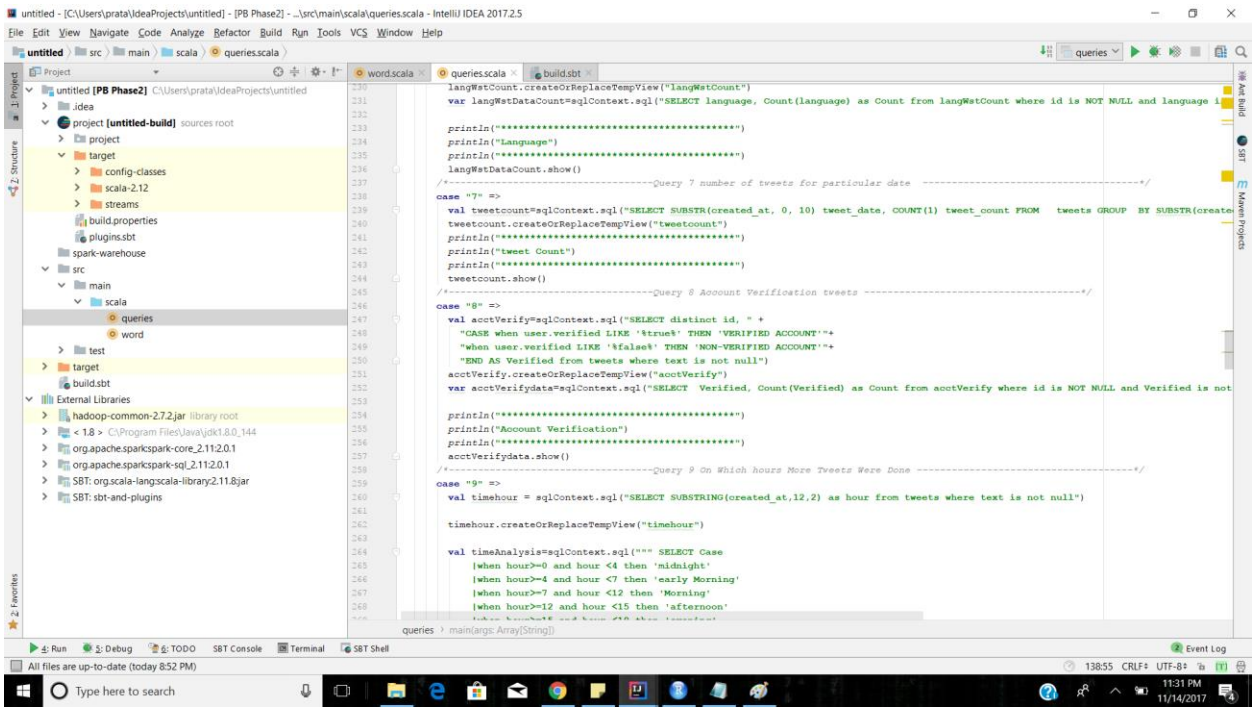
Query 7: Query for fetching Number of tweets for date.

Answer: This query is built to analyze the tweets – based on each individual date how many tweets are posted.

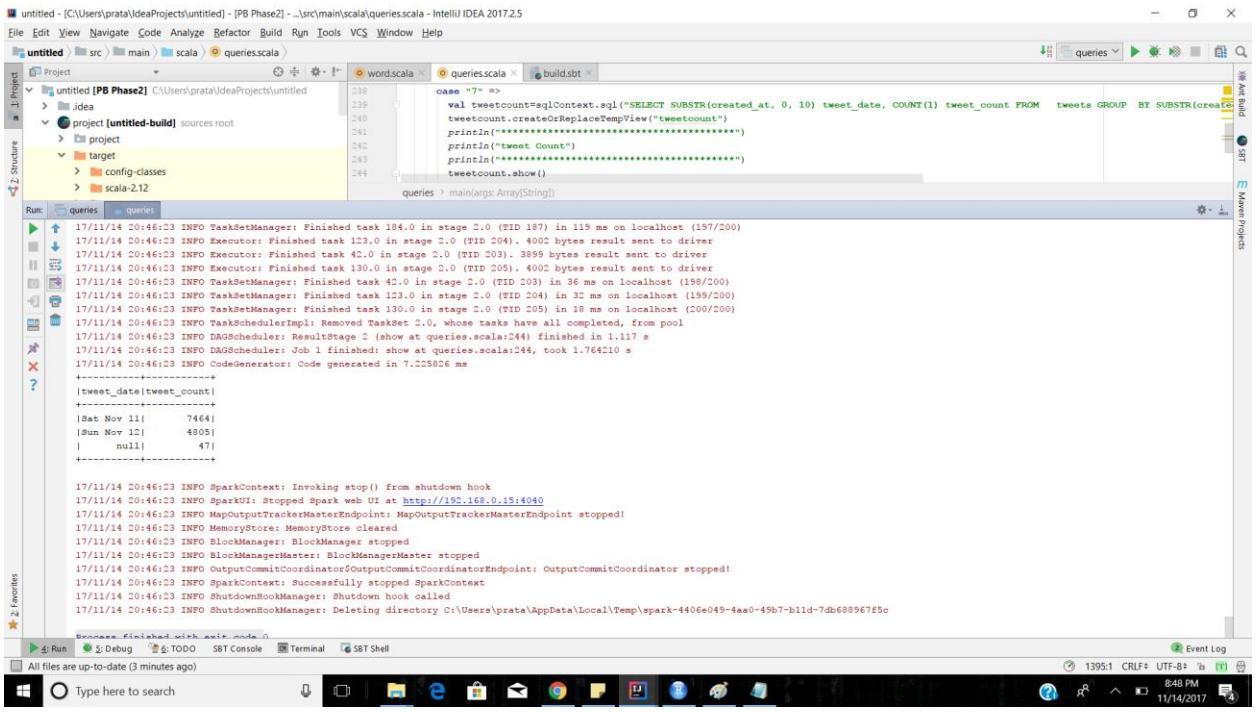
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to analyze, depending up on the collected data for each date how many tweets are posted .

Code:



Output:



Visualization:

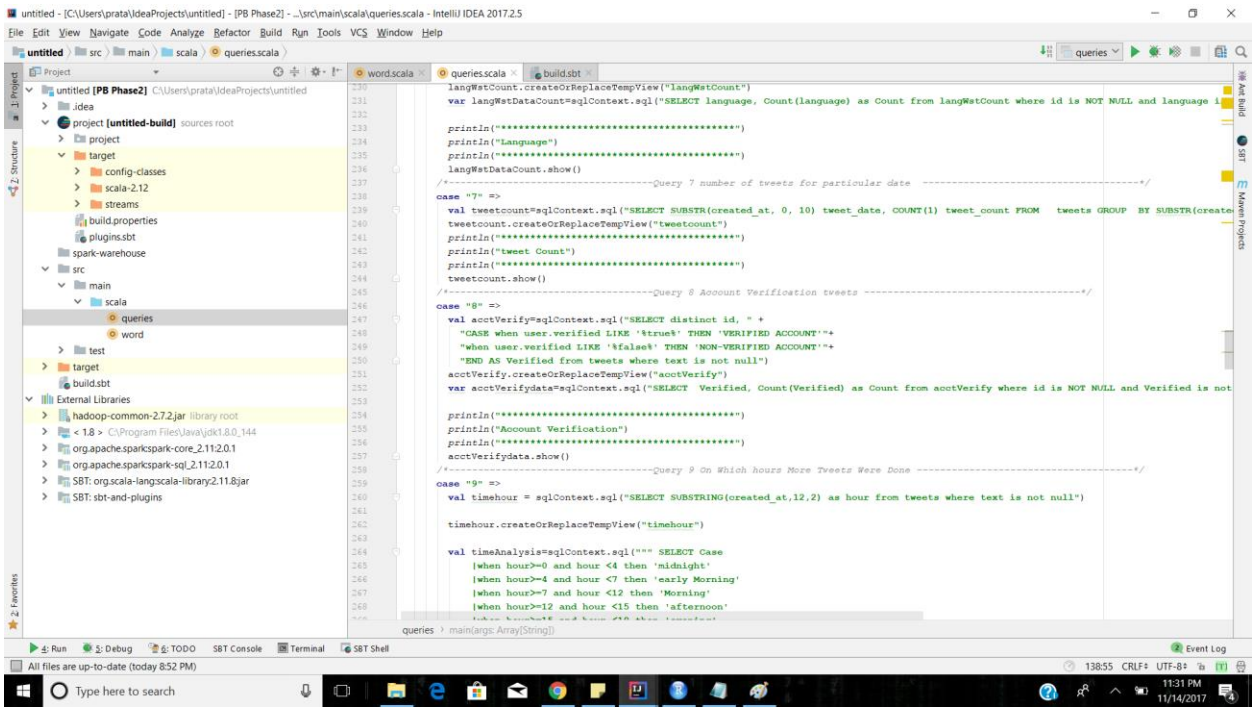
Query 8: Query for fetching Account Verified tweets.

Answer: This query is built to analyze the tweets – based on account verified tweets

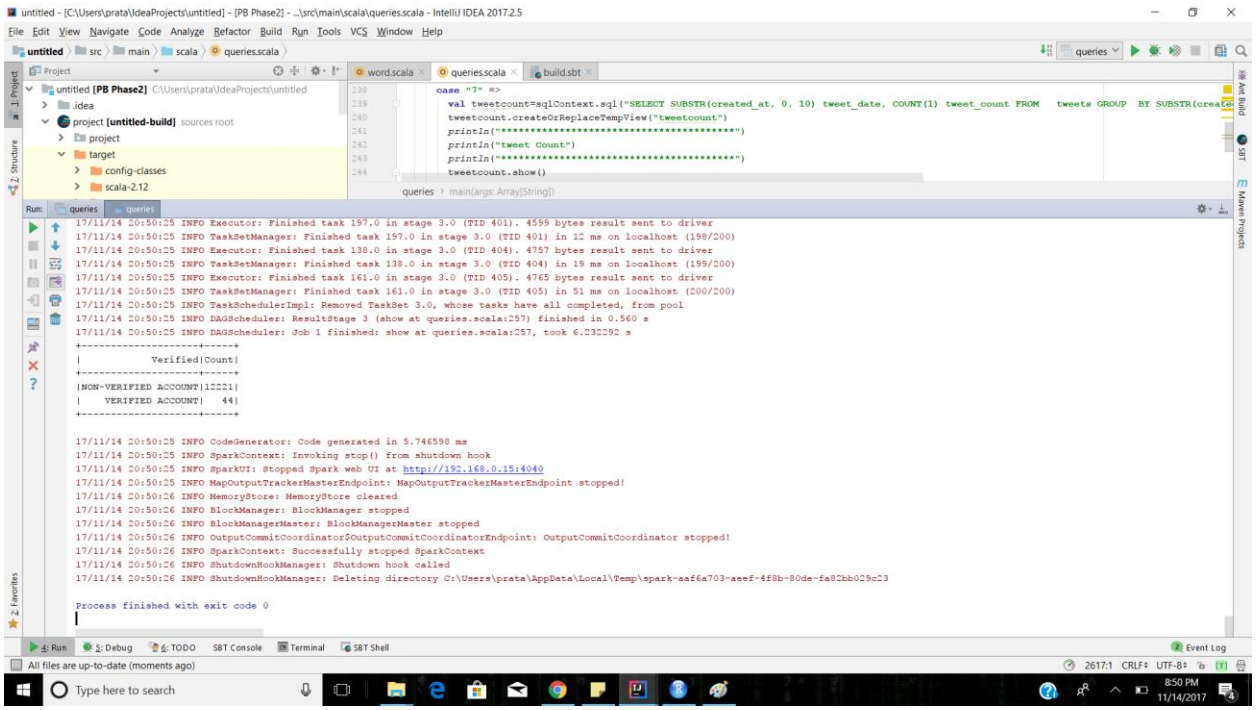
```
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','
#BlackBerry','#HTC
```

Such that built a query to analyze how many users are verified users. The accounts mainly differentiated by official check. Thus, counting the tweets which are posted from official accounts.

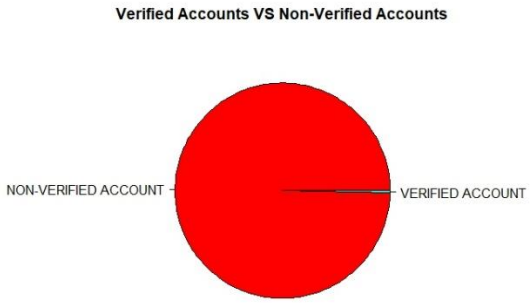
Code:



Output:



Visualization:



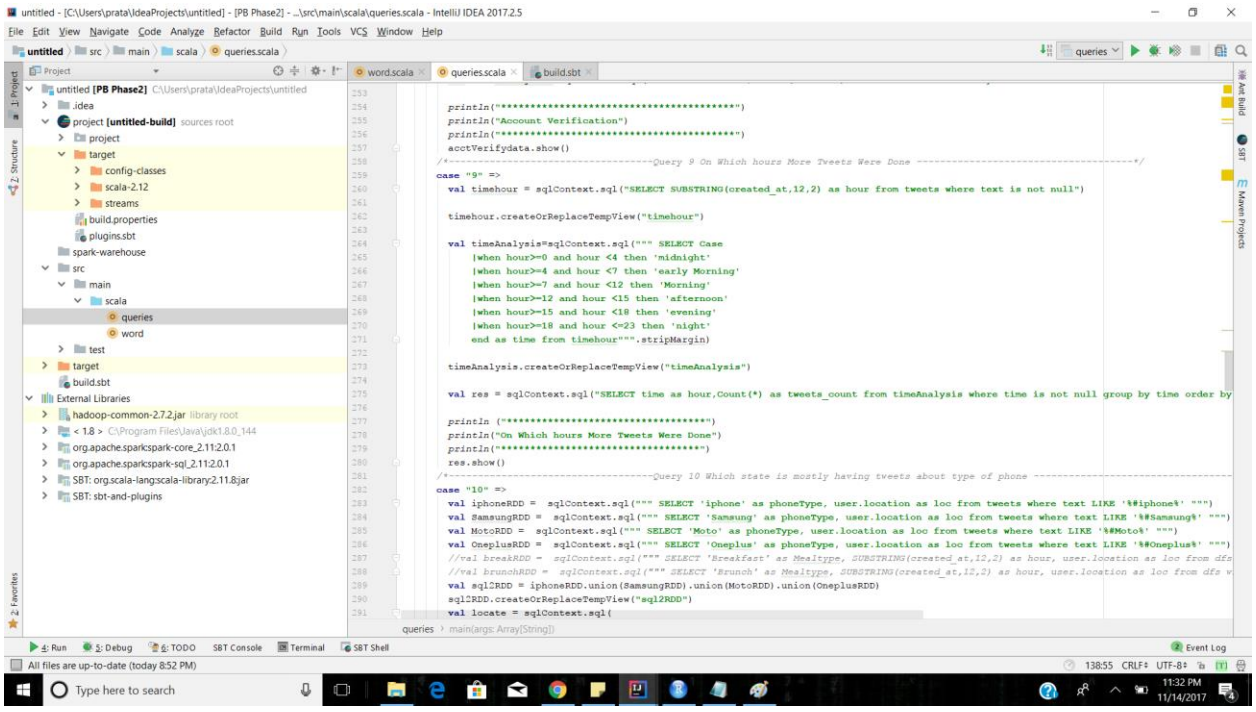
Query 9: Query for fetching On Which Hours More Tweets Were Done.

Answer: This query is built to analyze the tweets – based on hours of the tweets.

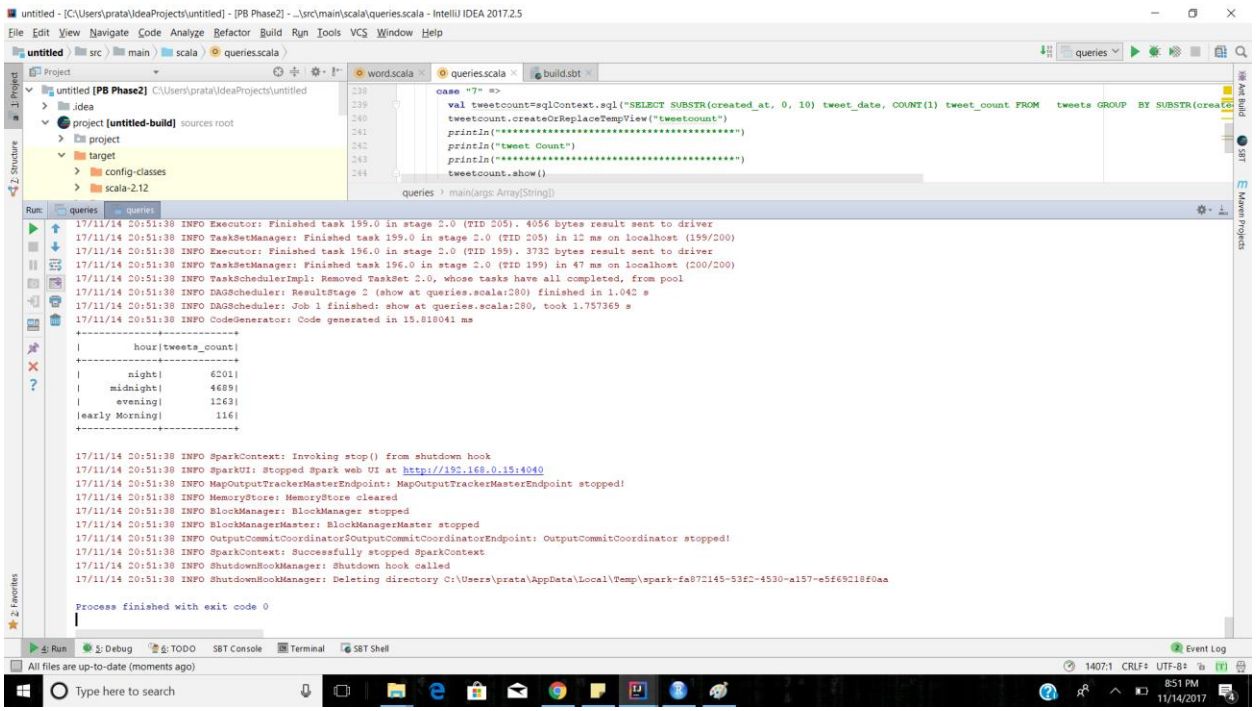
As we collected the data related to mobile phones
#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','#BlackBerry','#HTC

Such that built a query to analyze when the tweets are tweeted like-Morning, afternoon, night. So ,it results the time analysis of tweets

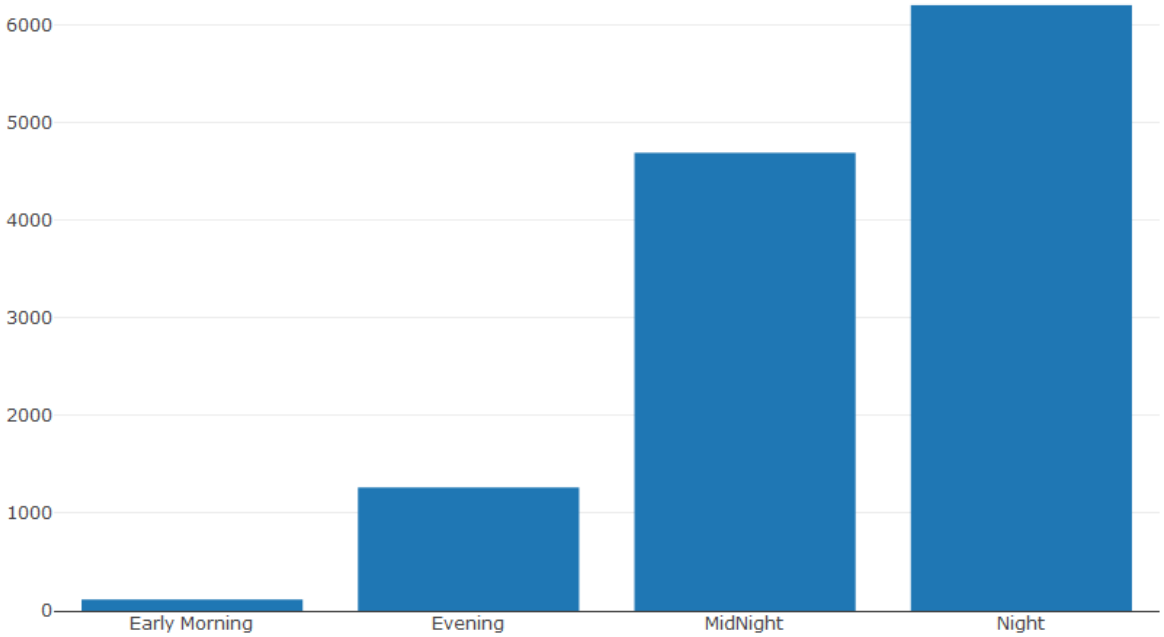
Code:



Output:



Visualization:



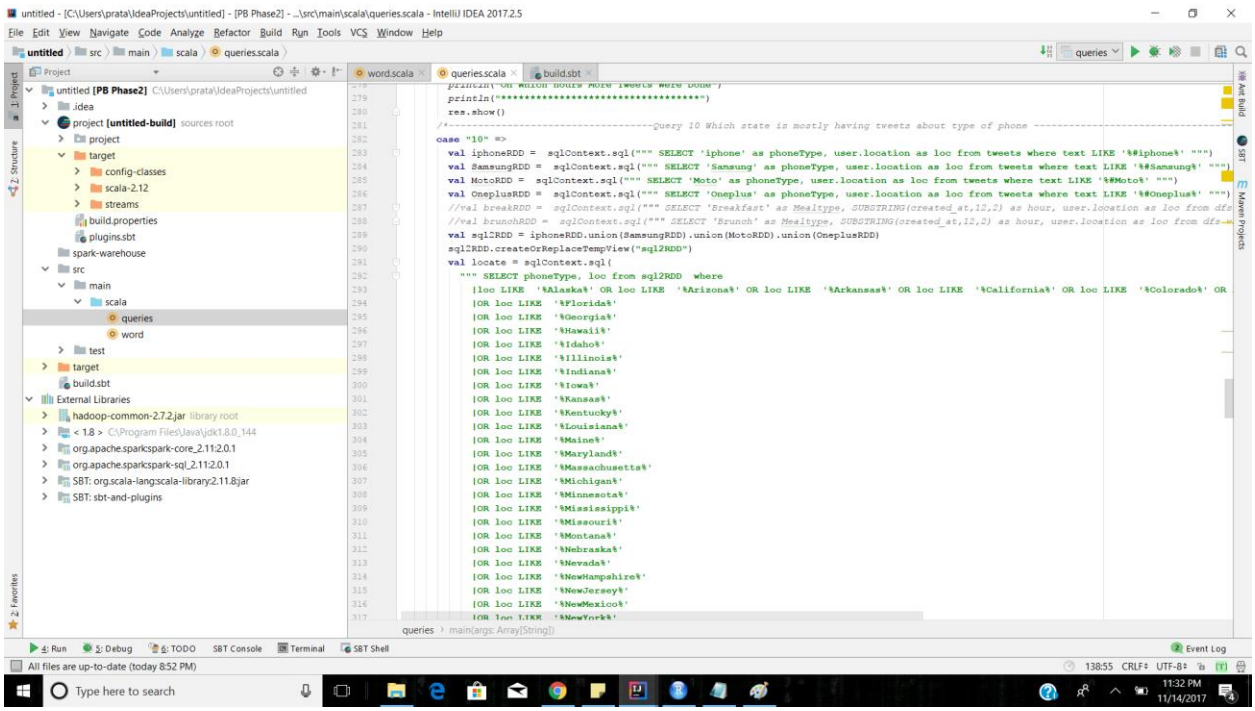
Query 10: Query for fetching Which state is mostly having tweets about type of phone.

Answer: This query is built to analyze the tweets – **Which state is mostly having tweets about type of phone**

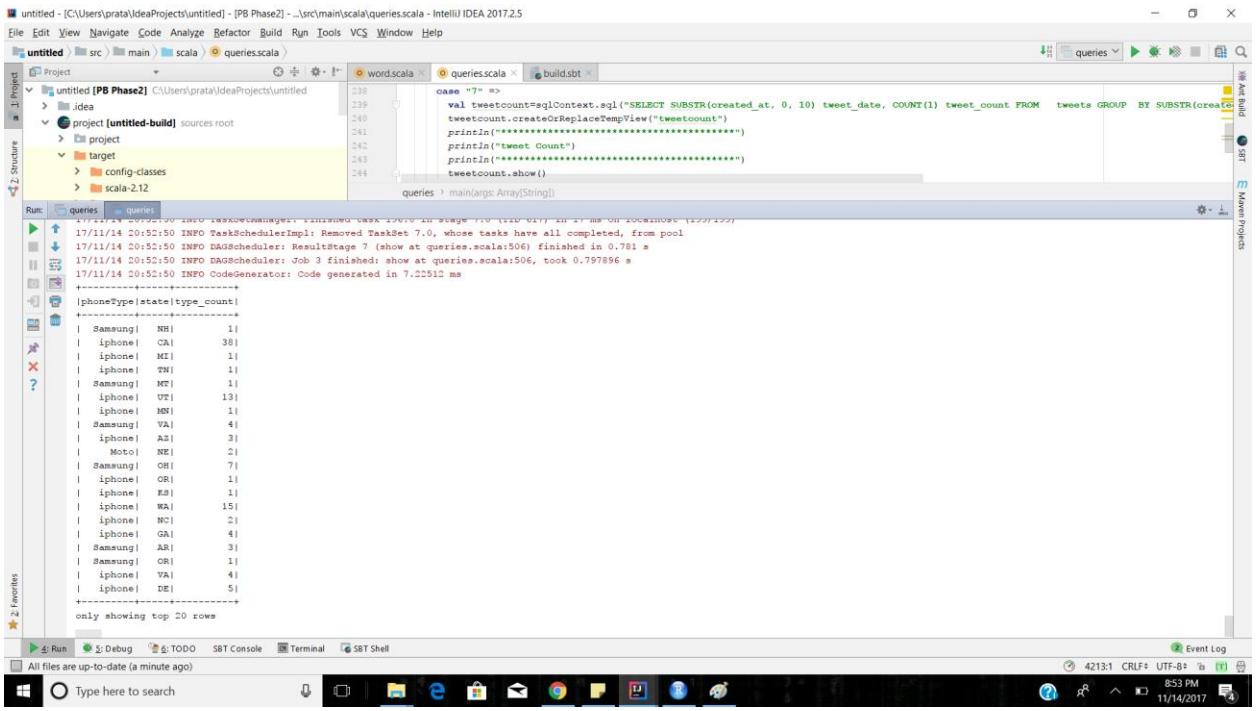
As we collected the data related to mobile phones **#iphone','#Samsung','#Moto','#Redmi','#Xiaomi','#Nokia','#lenovo','#oppo','#OnePlus','#BlackBerry','#HTC**

Such that built a query to perform

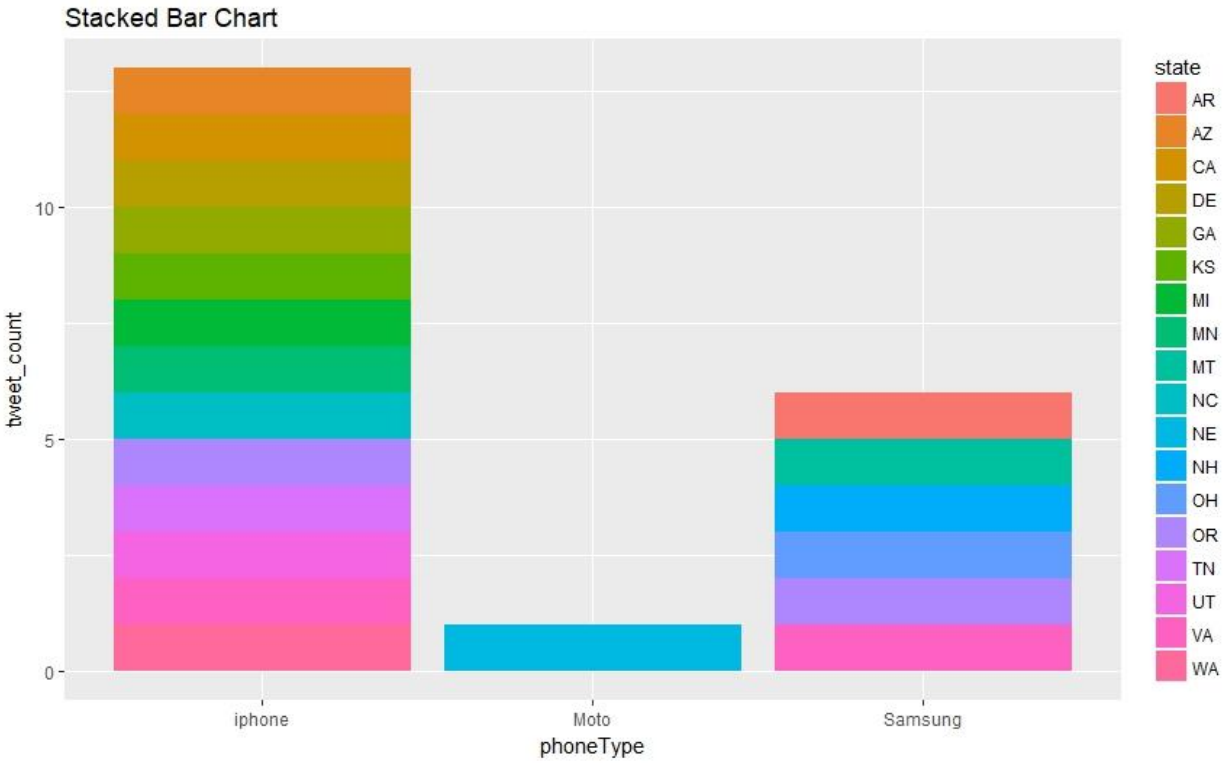
Code:



Output:



Visualization:



Source Code output log files link (Shared via drive)

https://drive.google.com/open?id=1YjOF_-tB8eqBvxPjePwLqpeDTFspDw5A