A Time-of-Use Analysis of Twitter Data

University of St. Thomas – St. Paul, MN
SEIS 736: Big Data Architecture
Stephen Ricci [ID: 101008319]

Introduction

The scope of this project is to analyze Twitter data, specifically focusing on the day and time of each tweet. The goal of this project was to isolate the day and hour of each tweet, with the intent to visualize user activity throughout each day of the week.

Data Source, Description and Schema

The data source utilized for this project was the 'tweets2' dataset that resides on the UST cluster. It is approximately 19 GB of Twitter data collected over the span of roughly two years, from September 2013 to March 2015. Upon collection, or shortly there afterwards, the Twitter JSON fields were flattened and five tab-separated fields were isolated. Those fields include: username, the actual tweet text, language indicator, date field ('day of the week' 'month' 'date' 'hh:mm:ss' 'time zone' 'year'), and a field containing NULL values. Please see example, below:

```
[CareerProGlobal	MyPeopleBiz Blog: Maximise your recruiting power with career sites
http://t.co/Ig88qohOoW	en	Thu Sep 26 10:13:00 CDT 2013	null]
```

Data Pre-Processing

Prior to performing the proposed analysis, the Twitter data was filtered to ensure that each tweet contained the expected, five tab-separated fields. After the initial filtering was complete, the date field was isolated, and a two-stage filtering process was conducted. The first stage of filtering performed on the isolated date field was to ensure that there were not any empty/missing values, therein. The second stage of filtering focused specifically on the time sub-field, and was performed to ensure that it met the expected length of eight characters (hh:mm:ss). After all the filtering was complete, the day of the week and hour of the day were isolated from the date field.

Spark Algorithm

The Spark algorithm utilized in this analysis is as follows:

1) Read in target file: /SEIS736/tweets2/allTweets
2) Map each record to an array, splitting each element by \t
3) Filter each tweet to ensure there are the five expected elements in each array
4) Isolate the date element from each tweet array
5) Map each date field to an array, splitting each element separated by a space (" ")
6) Filter each date array to ensure there are no empty/missing elements
7) Filter each time element of the date array to ensure that it is the expected eight characters in length
8) Isolate and map the day of the week and hour of the day from the date array, and convert the hour of the day element to an integer
9) Convert the day of the week/hour of the day RDD to a Spark Dataframe
10) Group by day of the week and hour of the day, and perform a count operation.
11) Write the results of the pervious step to .csv file for further analysis/visualization

Additional Tools/Ecosystem

Upon creating the dataframe used to perform the proposed analysis, it was initially saved as a .parquet file, and Hive was used to run queries against the dataframe contents. Ultimately, it proved more efficient, and just as effective, to run query operations directly against the dataframe in Spark. The results of the dataframe operations were written to a .csv file, and copied to the local file system from HDFS. WinSCP was then utilized to bring the .csv file from the local file system on the cluster to the local disk (C: Drive) on the laptop used throughout the project. Once written to the C: Drive, the contents of the .csv file were moved to an Excel document. From there, Tableau was used to connect to the Excel workbook and construct the data visualizations that are subsequently presented, herein.

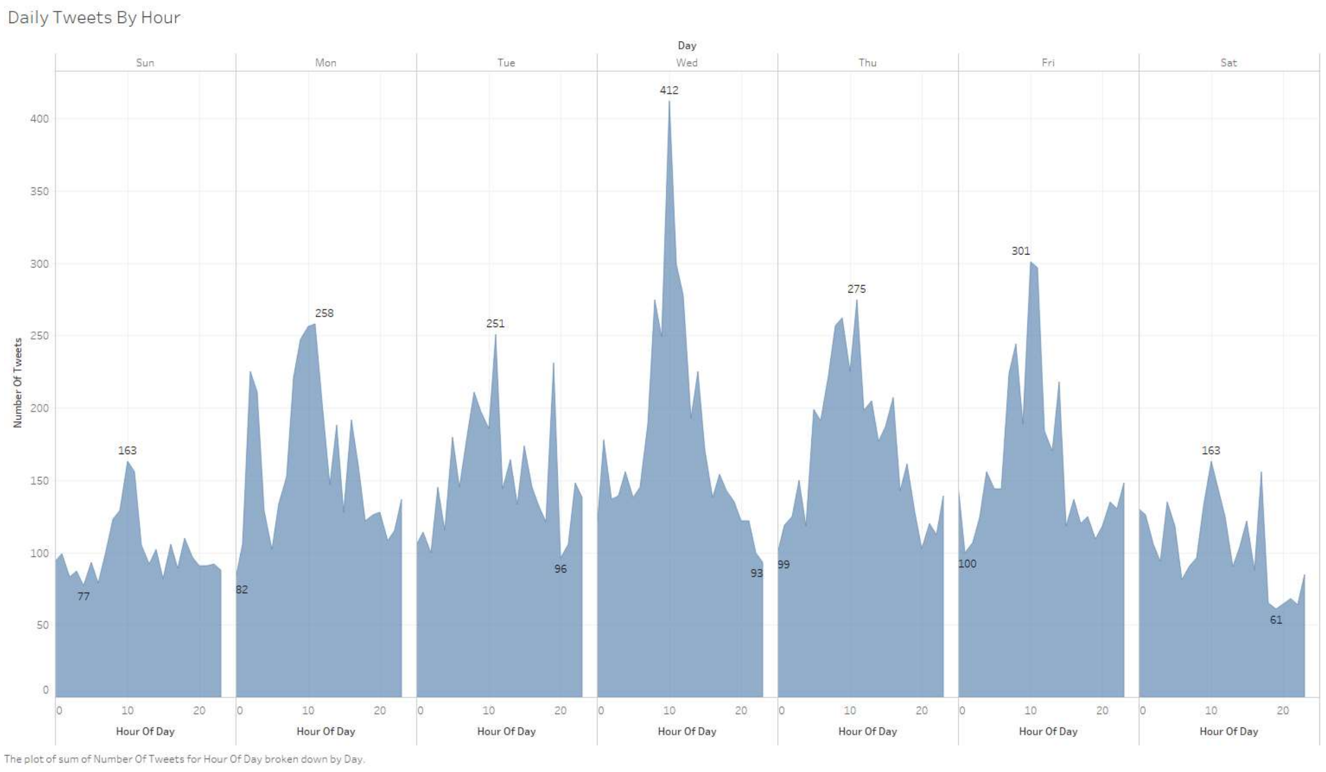
Bad Data Issues

While conducting the analysis of the Twitter dataset utilized in the project, there were bad data issues encountered, as one expects when analyzing large datasets. Overall the dataset consisted of reliable data; however, the bad data issues that did arise were attributed to missing/empty values. This is the very reason two filter operations are implemented on the isolated date field. Prior to the implementation of these filter operations, an ArrayOutOfBounds error would be thrown every time there was an attempt to conduct any of the dataframe operations used to perform the desired analysis. Thankfully, with some guidance from Dr. Rubin, the root cause was accurately identified and the missing/empty values were filtered out, which allowed for the desired analysis to be conducted.

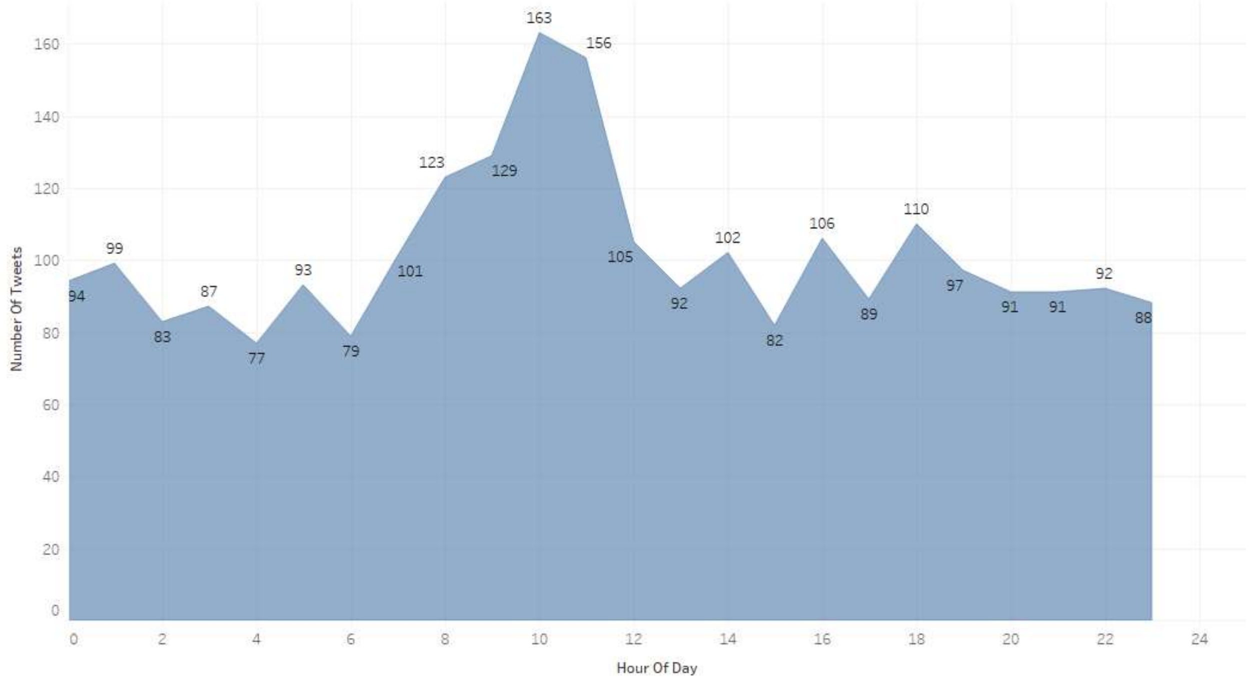
Another interesting data issue was identified in the results once the analysis was conducted. That is, an extraordinarily substantial number of tweets were counted at 20:00 (8:00 PM) for all of the Saturday tweets captured in the dataset. So much so that the data point was removed from visualization of the data all together, as it skewed the results dramatically. In the condensed time available to perform the analysis, since the original project plan did not work out, there was not sufficient time to fully investigate what may be causing the anomaly. One possible explanation could be that an issue occurred when collecting the dataset, or when the JSON fields were flattened to extract the five fields present for each tweet in the file. Of course, another explanation for the strange number of tweets collected at 8:00 PM on Saturdays throughout the dataset could be that there was an error when filtering/isolating the date field from the tweets. However, if that was the case, it seems plausible that abnormally high numbers of tweets would be observed at other dates and times throughout the obtained results.

Analysis Output

The analysis yielded an output fairly consistent with what was expected. That being a higher number of tweets sent during the middle of the day, when the majority of people are active. The higher number of tweets during the middle of the day may also be correlated to the fact that a large number of people take a lunch break, and could potentially use that time to contribute to the Twitter universe – something that would be interesting to investigate further. An observed result that was not expected was the fact that Wednesdays, mid-day show the highest number of tweets be made by users. Please see the visualizations of the obtained results, below:

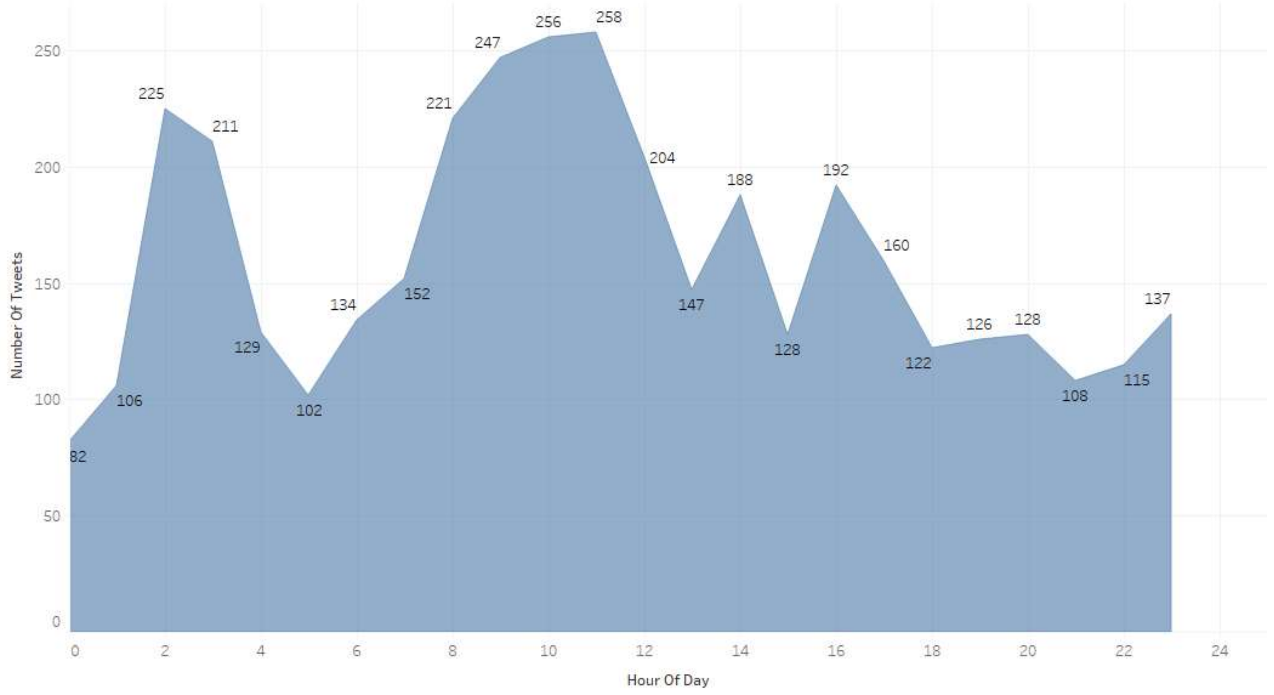


Sunday: Tweet By Hour



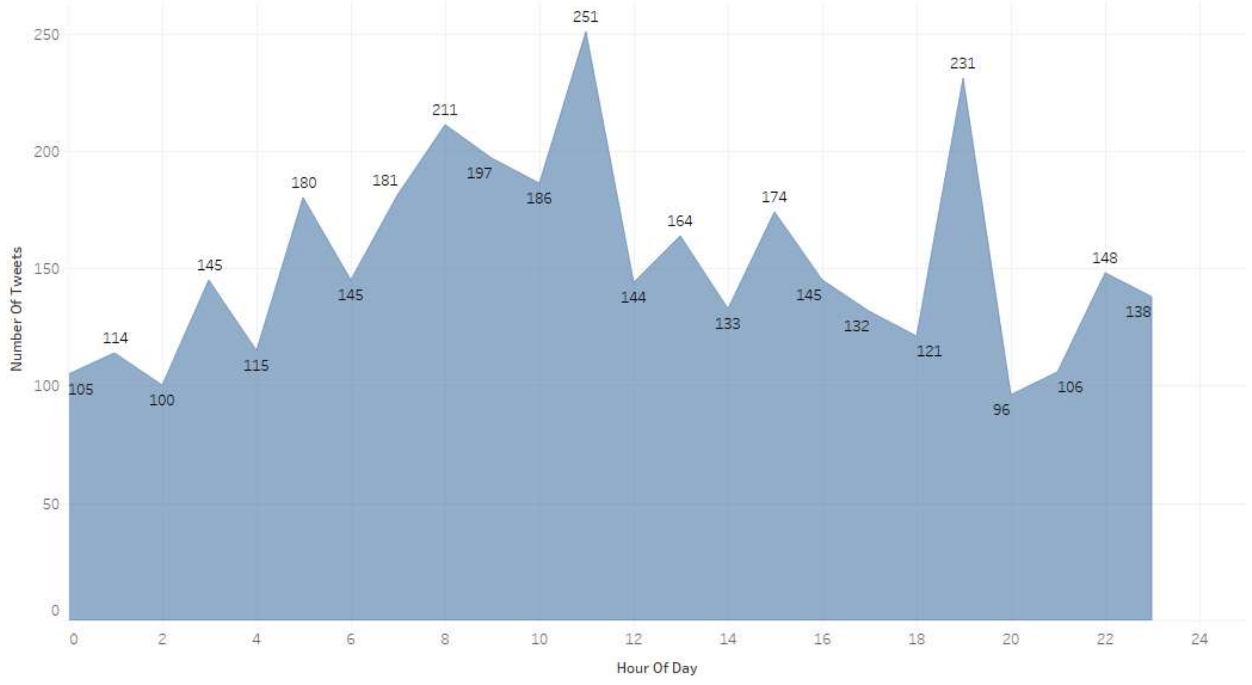
The plot of sum of Number Of Tweets for Hour Of Day.

Monday: Tweets By Hour



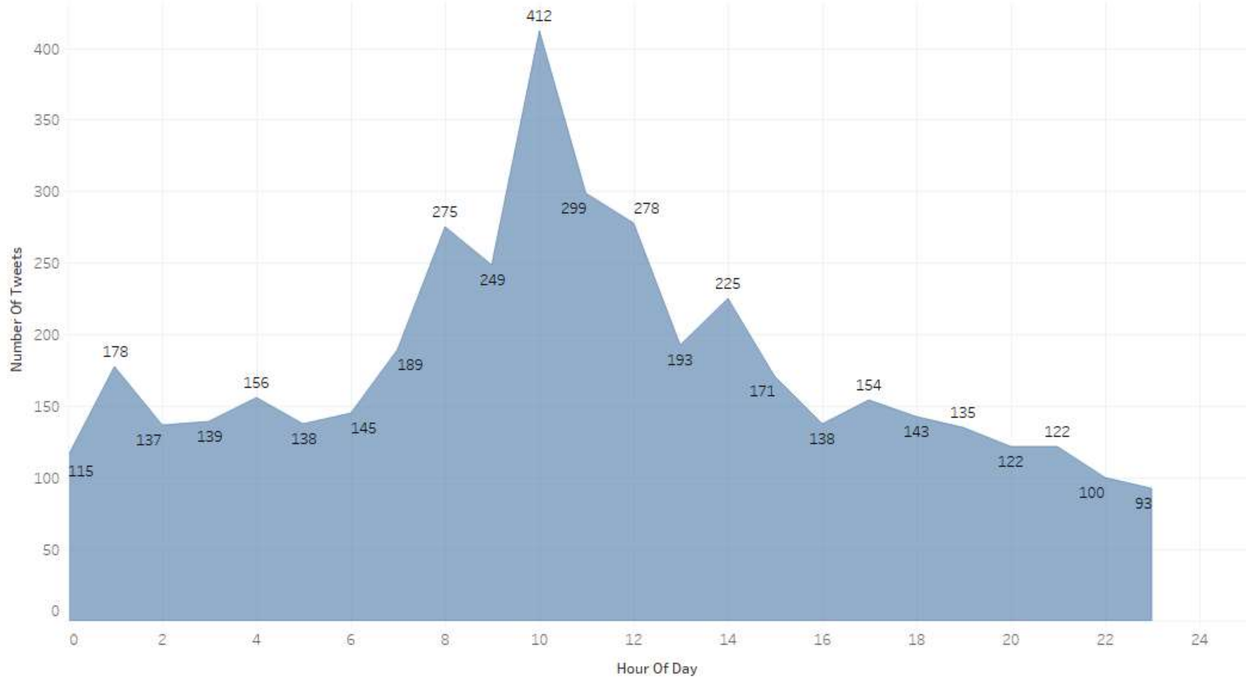
The plot of sum of Number Of Tweets for Hour Of Day.

Tuesday: Tweets By Hour



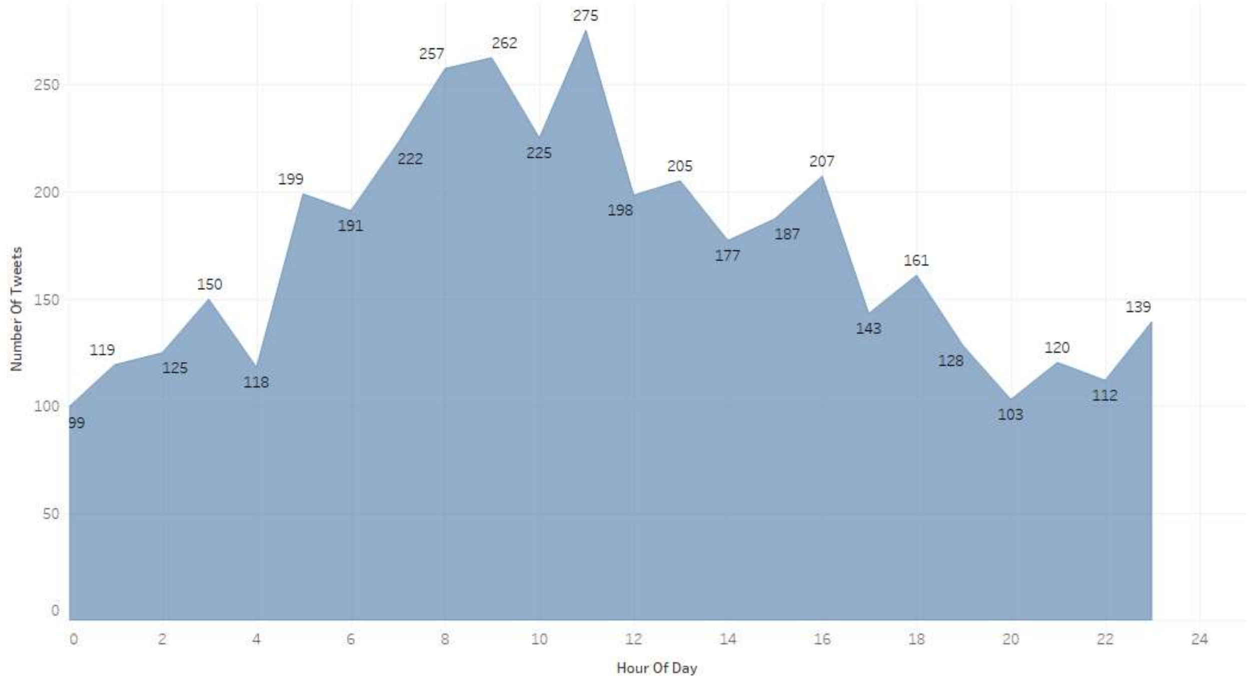
The plot of sum of Number Of Tweets for Hour Of Day.

Wednesday: Tweets By Hour



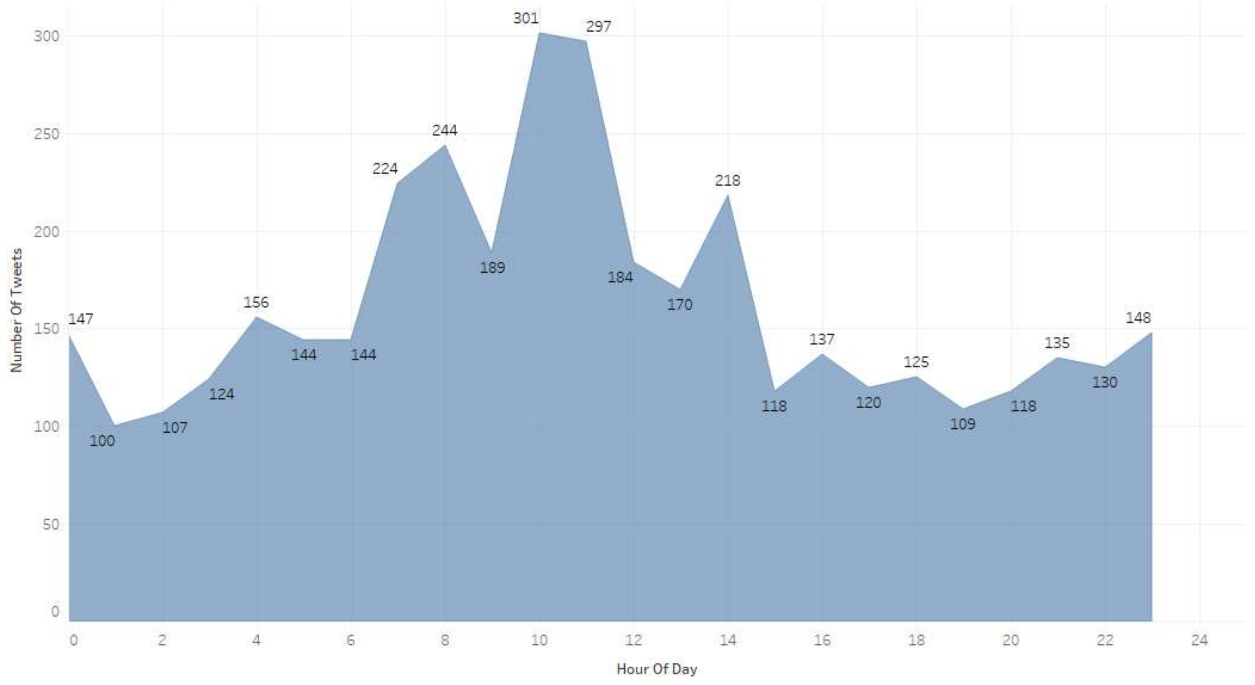
The plot of sum of Number Of Tweets for Hour Of Day.

Thursday: Tweets By Hour

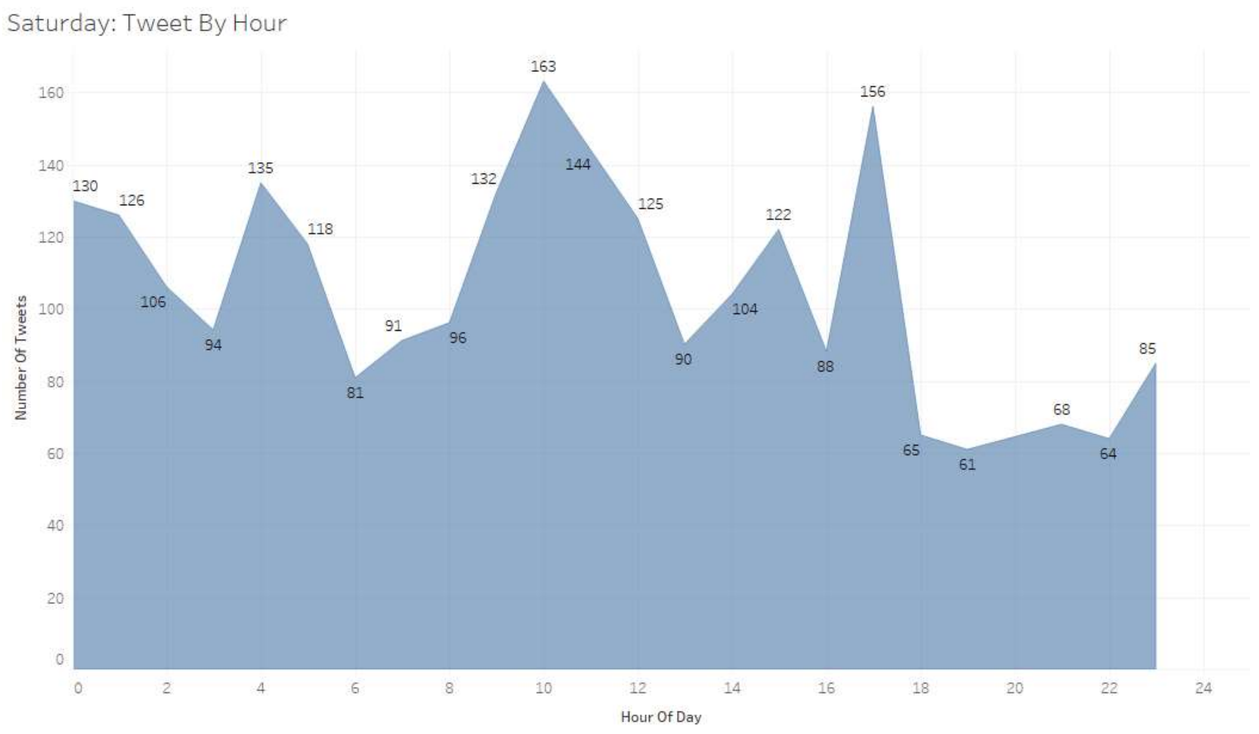


The plot of sum of Number Of Tweets for Hour Of Day.

Friday: Tweets By Hour



The plot of sum of Number Of Tweets for Hour Of Day.



The plot of sum of Number Of Tweets for Hour Of Day.

**Data point for 20:00 has been removed due to anomalous number of tweets

Performance/Scale Characteristics

All the code and operations conducted on the target /SEIS736/tweets2 dataset were first run on the smaller /SEIS736/tweets dataset to ensure that everything was working as designed. On the smaller dataset, the performance was great – extremely fast. In terms of scalability, all the code/operations, when applied to the larger, target dataset, performed as expected. That is, a little slower, but the performance was still very fast for processing the amount of data in the /SEIS736/tweets2 dataset. Utilizing Spark dataframes made conducting the desired analysis very efficient, and much easier than what was expected prior to the start of the project.

Conclusions

Spark is an amazing tool for processing big data, and having the opportunity to utilize it in this capacity has definitely sparked my interest (pun intended) in continuing to focus on developing my skills with the goal of utilizing it in my current position, and ultimately pursue new career opportunities. Furthermore, starting to learn to program in Scala has been great, and something I will also continue to practice, going forward.

What would I have done differently, if I had to do this all over again? To start, I will give the cliché answer of: start working on my project much sooner than I did. However, in that same breath, I will also say that my wife having a baby and moving, back-to-back weeks, in the middle of the semester did not boost my ability to devote the time and attention that was needed at the critical starting point of the project. The effects those life-changing events ended up having on my ability to focus on/excel in my studies were far greater than expected, and looking back, knowing what I know now, I would have started my project the first week of class. I would also change the initial scope of my project, and instead of attempting to collect a dataset to analyze, I would have utilized a pre-existing dataset from the very beginning. Not being able to correctly implement data collection methods was a real detriment to my project, overall. In hindsight, I should have made the decision to move away from trying to collect a dataset much sooner than I did, which would, in turn, have given me an opportunity to better analyze the dataset I did end up using, as well as address the data issues that did ultimately arise.

Code and Commands

Scala Code

```
spark-shell --packages com.databricks:spark-csv_2.10:1.5.0

//read in dataset
scala> val tweets = sc.textFile("/SEIS736/tweets2/allTweets")

//map tweets - split by tab
scala> val splitTweet = tweets.map(x => x.split("\\t"))

//filter tweets to ensure there are 5 elements per tweet array
scala> val filterTweet = splitTweet.filter(x => x.size == 5)

//isolate date element from tweet
scala> val dates = filterTweet.map(x => x(3))

//map date elements - split by " "
scala> val dateMap = dates.map(x => x.split(" "))

//filter date array elements to ensure there are no empty/missing elements
scala> val filterDateMap = dateMap.filter(x => x.nonEmpty)

//filter time element of the date arrary to ensure all are 8 characters in length
scala> val dblFilter = filterDateMap.filter(x => (x(3).length == 8))

//isolate the dayOfTheWeek and hourOfDay - convert the hourOfDay to Int
scala> val dayTime = dblFilter.map(x => (x(0),x(3).substring(0,2).toInt))

//print three lines to verify what fields have been isolated
scala> dayTime.take(3).foreach(println)

//create tweet dataframe
scala> val tweetDF = dayTime.toDF("Day", "HourOfDay")

//verify dataframe contents (top 20 rows)
scala> tweetDF.show

//perform groupBy and count operations on dataframe and write results to csv file
 scala> tweetDF.groupBy("Day","HourOfDay").count.write.format("com.databricks.spark.csv").option("
header" ,"true").save("/user/ricci/dayHourTweetCount")

//write dataframe to parquet file
scala> val o = Map("path" -> "/user/ricci/finalProject")
scala> tweetDF.write.format("parquet").options(o).saveAsTable("fp")
```

Hadoop Commands

```
$hadoop fs -ls /user/ricci
```

```
$hadoop fs -copyToLocal /user/ricci/dayHourTweetCount /home/ricci/dayHourTweetCount
```