IS622 Week6 SparkR Confirmation and Stream Example

James Quacinella

10/10/2015

Initial Setup

```
# Setup SparkR
Sys.setenv(SPARK_HOME="/home/james/Software/spark-1.4.1-bin-hadoop2.6")
library(SparkR, lib.loc = "/home/james/Software/spark-1.4.1-bin-hadoop2.6/R/lib")
```

```
##
## Attaching package: 'SparkR'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, sample, table
```

```
sc <- sparkR.init()
sqlContext <- sparkRSQL.init(sc)
```

Week 6 Description

> WK 6 Mini Project: Streaming Data - The focus of this week is to implement the plan from week 5. Provide working code that samples, filters (removes), and counts the occurrence of specific elements in the data. Is the approach different from your original plan? Why or why not? Perform a simple analysis of your resulting data. Things of interest can include a histogram of counts over different windows, comparison of counts for different elements collected, comparing the behavior of counts versus what was not sampled, etc. The point is to be creative – identify some questions that you are curious about and investigate.

Much of the code here will be the same, as some of the counting was already done. The code below will expand on things, plot some of the username and hashtag counts. I would have liked to filter tweets down by timestamp, and maybe comparing counts during different times of the day. Sadly, I am running out of time for this assignment, but doing so would have meant some extra code in my filtering function to include the tweet timestamp (maybe just the hour), produce another column in the dataframe for the timestamp, and then used a Spark function to group things not only based on hashtag or username, but the hour number as well.

Week 6 Setup

Lets setup the twitteR module:

```
library(twitteR)
library(streamR)
library(ggplot2)
library(stringr)

# I know this is bad form
setup_twitter_oauth('yBMKyokT0Io0g9sXnCJ6ZZyYB',
                    'B9v58Sm06hRtHpYYoHpFVGb5BEpAUWAORIPumqfMMdM7NwemX4',
                    '16562593-mxuDgZWbfnT4Nxdq7gXQe3K1HkRrw8PWkzQpOZsjp',
                    'qaafVHWbQMlkZ97V9wIE8o7pJwQObIS91blJHYEjCwMZd')
```

```
## [1] "Using direct authentication"
```

```
# # From http://bogdanrau.com/blog/collecting-tweets-using-r-and-the-twitter-streaming-api/
# library(ROAuth)
# requestURL <- "https://api.twitter.com/oauth/request_token"
# accessURL <- "https://api.twitter.com/oauth/access_token"
# authURL <- "https://api.twitter.com/oauth/authorize"
# consumerKey <- "yBMKyokT0Io0g9sXnCJ6ZZyYB" # From dev.twitter.com
# consumerSecret <- "B9v58Sm06hRtHpYYoHpFVGb5BEpAUWAORIPumqfMMdM7NwemX4"
#
# my_oauth <- OAuthFactory$new(consumerKey = consumerKey,
#                              consumerSecret = consumerSecret,
#                              requestURL = requestURL,
#                              accessURL = accessURL,
#                              authURL = authURL)
#
# my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl")
# save(my_oauth, file = "my_oauth.Rdata")
```

Streaming Data From Twitter

First we load some tweets from the #politics hashtag, to then be processed for what usernames each tweet mentions. We store all these names in a new dataframe, and then submit that dataframe to spark:

```
# filterStream(file.name = "tweets.json", # Save tweets in a json file
#              track = c("#politics"),
#              language = "en",
#              timeout = 120,
#              oauth = my_oauth) # Use my_oauth file as the OAuth credentials
#
# tweets.df <- parseTweets("tweets.json", simplify = FALSE)

# Grab tweets and serialize to disk for future use
if(file.exists("tweets.Robj")) {
  load("tweets.Robj")
} else {
  tweets <- searchTwitter("#politics", n=2000)
  save(tweets, file="tweets.Robj")
}
```

```r
# Function to take in tweet row and return list of usernames mentioned in tweet
getUsernames <- function(tweet) {
  content <- tweet$text
  usernames <- unlist(lapply(unlist(strsplit(content, " ")),
                             function(word) {
                               if (substr(word,1,1) == "@" & try(nchar(word)) > 1) {
                                 return(tolower(gsub("[[:punct:]]", "", word)))
                               }
                             }))
  usernames <- usernames[ !is.null(usernames) ]
  usernames
}

# Function to take in tweet row and return list of hashtags mentioned in tweet
# Same as above really, but filtering source hashtags (not sure why its not working)
getHashtags <- function(tweet) {
  content <- tweet$text
  hashtags <- unlist(lapply(unlist(strsplit(content, " ")),
                             function(word) {
                              word <- tolower(word);
                               if (substr(word,1,1) == "#" & word != "#politics" & try(nchar(word)) > 1
                                 return(tolower(gsub("[[:punct:]]", "", word)))
                               }
                             }))
  hashtags <- hashtags[ !is.null(hashtags) ]
  hashtags
}

# Pull out user names from tweets
usernames <- unlist(lapply(tweets, getUsernames))
usernames.df <- data.frame(username=usernames)

# Pull out user names from tweets
hashtags <- unlist(lapply(tweets, getHashtags))
hashtags.df <- data.frame(hashtag=hashtags)

# Create spark data frame from this list of users
usernames.sdf <- createDataFrame(sqlContext, usernames.df)
hashtags.sdf <- createDataFrame(sqlContext, hashtags.df)
```

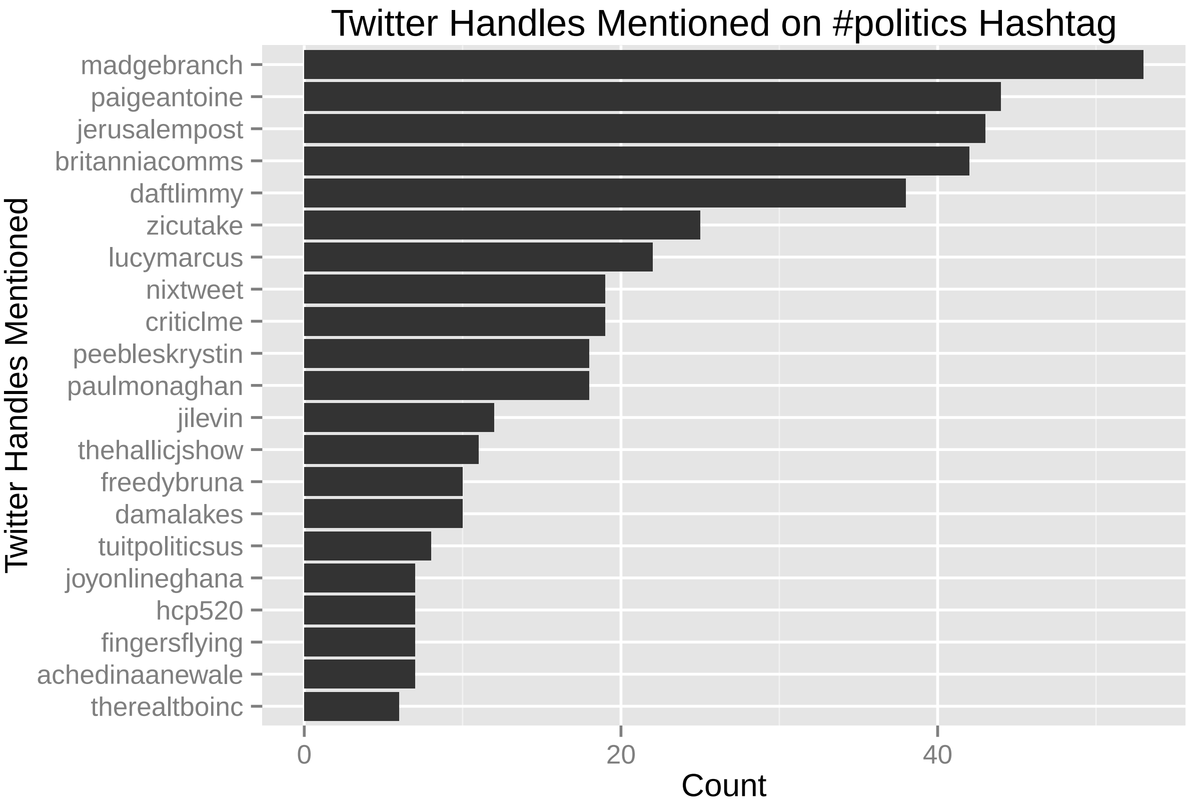
Username Results

Lets plot the distribution of usernames:

```r
# This URL helped understand these functions: https://spark.apache.org/docs/latest/sparkr.html
username_results <- summarize(group_by(usernames.sdf, usernames.sdf$username), counts=n(usernames.sdf$us
username_final_results <- collect(arrange(username_results, desc(username_results$counts)))

# Plot usernames mentioned
p<-ggplot(data=username_final_results[username_final_results$counts > 5, ], aes(x=reorder(username, cou
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Twitter Handles Mentioned on #politics Hashtag") +
  ylab("Count") + xlab("Twitter Handles Mentioned")
ggsave(filename="handles_on_politics.jpg", plot=p)
```

```
## Saving 6.5 x 4.5 in image
```

```
p
```



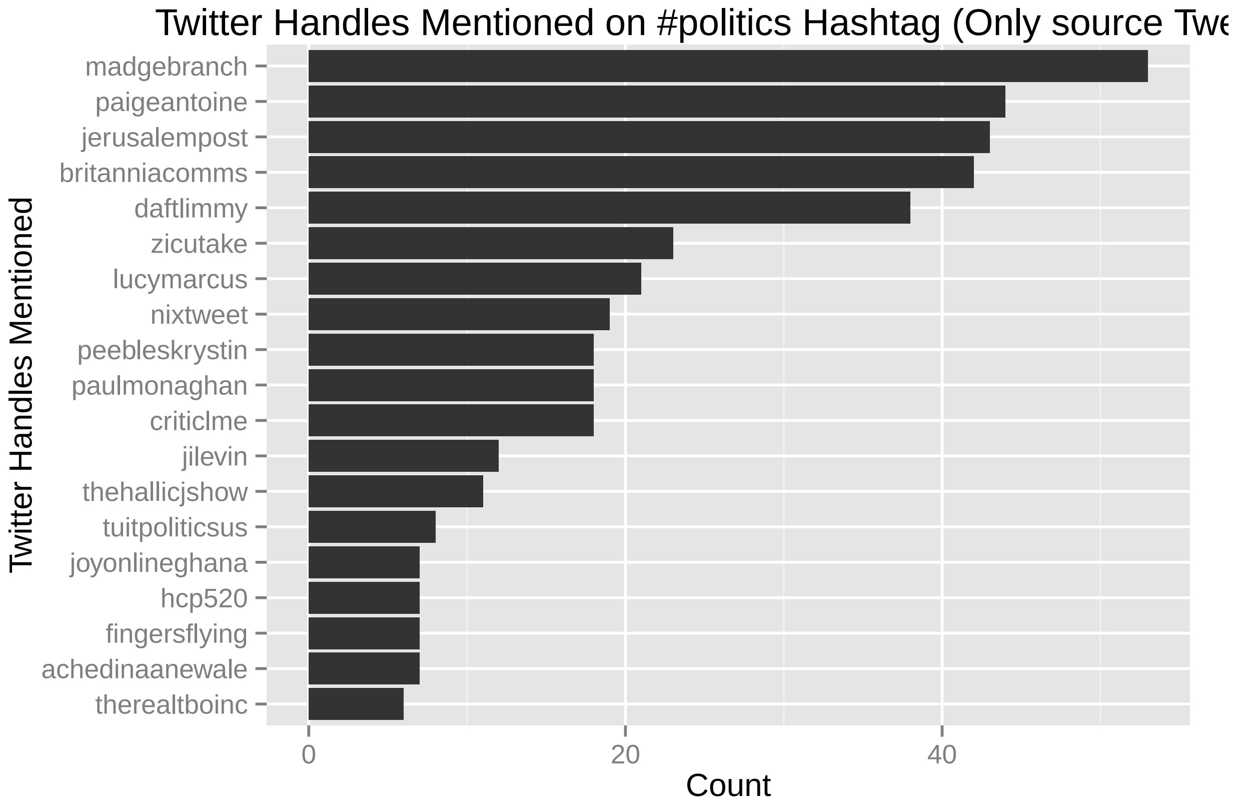
Some interesting results. The top username mentioned is the jerusalempost, which is a political publication. Some of the results are indicitive of spam, like amznfavorites, which looks to be a commerical account. These results indicate those usernames that are being mentioned very often on this hastag, which means they are either being mentioned a lot, or are in a lot of conversations.

A good idea for a filter here might be to filter out tweets that are replies. Lets redo the analysis with this in mind:

```
tweets.df <- twListToDF(tweets)
tweets.df <- tweets.df[is.na(tweets.df$replyToSN), ]

tweets_source <- list(tweets.df$text)

getUsernames2 <- function(tweet) {
  content <- tweet  # hack, since we are using diff data type now
  usernames <- unlist(lapply(unlist(strsplit(content, " ")),
                             function(word) {
                               if (substr(word,1,1) == "@" & try(nchar(word)) > 1) {
                                 return(tolower(gsub("[[:punct:]]", "", word)))
                               }
                             }))
  usernames <- usernames[ !is.null(usernames) ]
  usernames
}

usernames <- unlist(lapply(tweets_source, getUsernames2))
usernames.df <- data.frame(username=usernames)
usernames.sdf <- createDataFrame(sqlContext, usernames.df)

username_results <- summarize(group_by(usernames.sdf, usernames.sdf$username), counts=n(usernames.sdf$u
username_final_results <- collect(arrange(username_results, desc(username_results$counts)))

# Plot usernames mentioned
p <- ggplot(data=username_final_results[username_final_results$counts > 5, ], aes(x=reorder(username, c
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Twitter Handles Mentioned on #politics Hashtag (Only source Tweets)") +
  ylab("Count") + xlab("Twitter Handles Mentioned")
ggsave(filename="handles_filtered_on_politics.jpg", plot=p)
```

```
## Saving 6.5 x 4.5 in image
```

```
p
```



This ends up not telling us too much, since the filter is ineffective. It might have been on a larger set, but this did not help. We would need other methods to filter our ‘spammy’ usernames that are using a popular hashtag to push tweets that will get views.

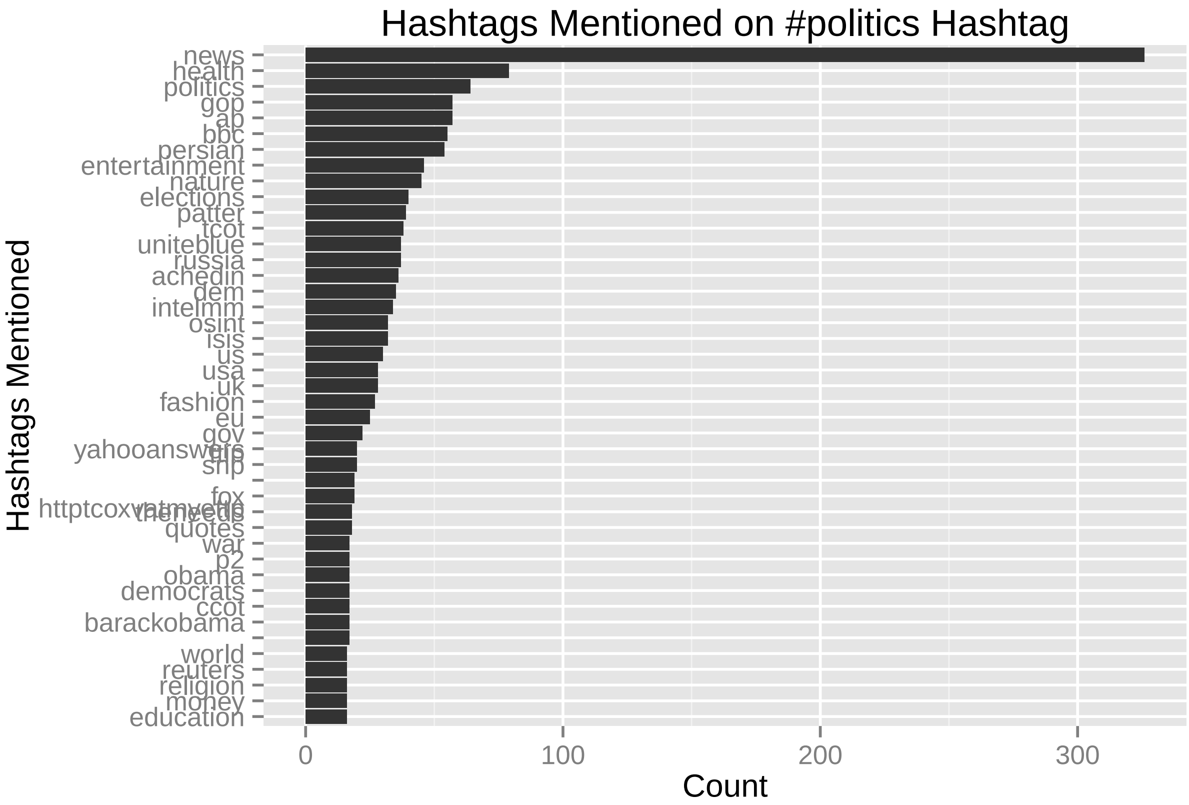
Hashtag Results

Lets count the number of hashtags seen in the stream:

```
hashtag_results <- summarize(group_by(hashtags.sdf, hashtags.sdf$hashtag), counts=n(hashtags.sdf$hashta
hashtag_final_results <- collect(arrange(hashtag_results, desc(hashtag_results$counts)))

# Plot hashtags mentioned
p <- ggplot(data=hashtag_final_results[hashtag_final_results$counts > 15, ], aes(x=reorder(hashtag, cou
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Hashtags Mentioned on #politics Hashtag") +
  ylab("Count") + xlab("Hashtags Mentioned")
ggsave(filename="hashtags_on_politics.jpg", plot=p)
```

```
## Saving 6.5 x 4.5 in image
```

```
p
```

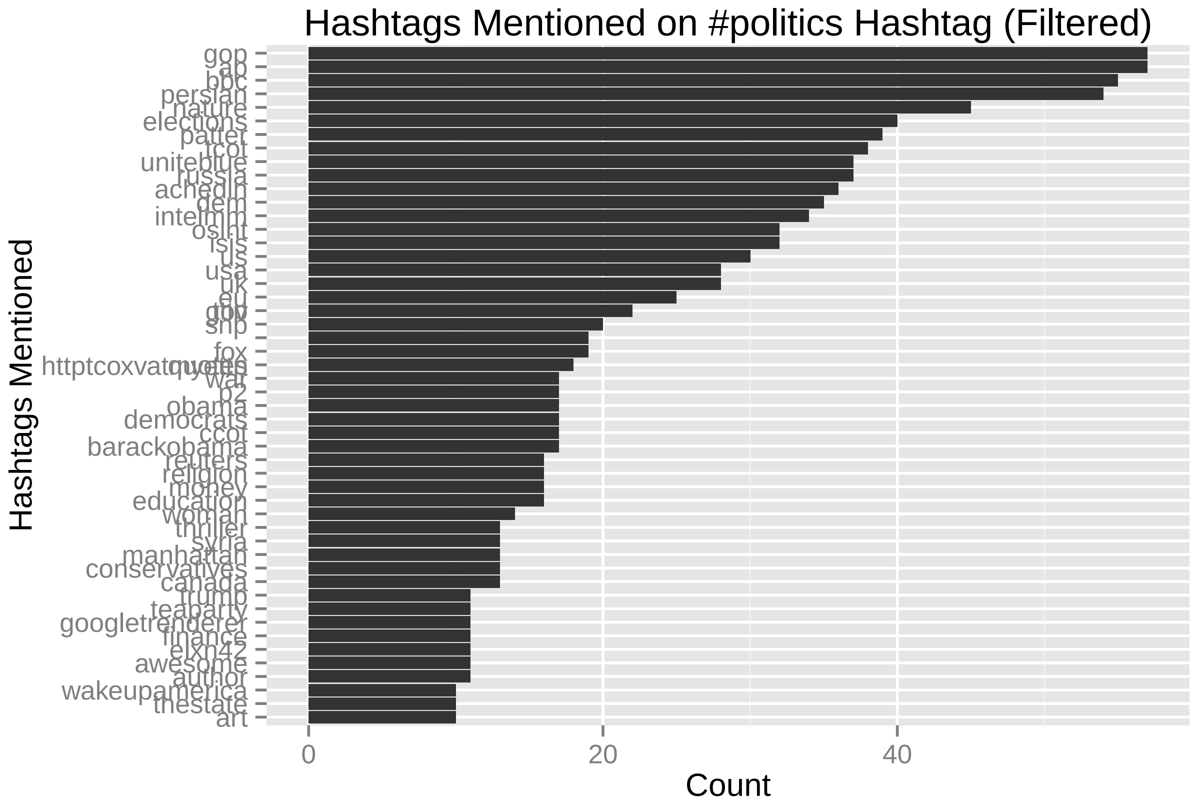


Some of these hastags are useless. Not sure why politics didn't get filtered out, but 'news', 'feedly' and 'health' are useless. I will redo the analysis below with some filtering of the data before putting it into spark:

```
stoplist <- c("politics", "news", "feedly", "health", "yahoonews", "yahooanswers", "entertainment", "fa
getHashtags2 <- function(tweet) {
  content <- tweet$text
  hashtags <- unlist(lapply(unlist(strsplit(content, " ")),
                            function(word) {
                                is_hashtag <- (substr(word,1,1) == "#")
                                word <- gsub("[[:punct:]]", "", tolower(word));
                                if (is_hashtag & !(word %in% stoplist) & try(nchar(word)) > 1) {
                                  return(word)
                                }
                              }))
  hashtags <- hashtags[ !is.null(hashtags) ]
  hashtags
}

hashtags <- unlist(lapply(tweets, getHashtags2))
hashtags.df <- data.frame(hashtag=hashtags)
hashtags.sdf <- createDataFrame(sqlContext, hashtags.df)
hashtag_results <- summarize(group_by(hashtags.sdf, hashtags.sdf$hashtag), counts=n(hashtags.sdf$hashtag
hashtag_final_results <- collect(arrange(hashtag_results, desc(hashtag_results$counts)))

# Plot hashtags mentioned
p <- ggplot(data=hashtag_final_results[1:50, ], aes(x=reorder(hashtag, counts), y=counts)) +
  geom_bar(stat="identity") +
  coord_flip() +
  ggtitle("Hashtags Mentioned on #politics Hashtag (Filtered)") +
  ylab("Count") + xlab("Hashtags Mentioned")
ggsave(filename="hashtags_filtered_on_politics.jpg", plot=p)
```

```
## Saving 6.5 x 4.5 in image
```

```
p
```



As you can see, we have some useful results. It seems like ‘conservative’ hashtags are near the top, with ‘gop’ and ‘tcot’ being amoung the top hashtags, with other tags being prevelant as well, like ‘trump’, ‘donaldtrump’, ‘teaparty’ and ‘conservatives’. Some other importat ones are ‘dem’, ‘uniteblue’, ‘obama’ and ‘democrats’. If we could monitor these counts by grouping by time first, we could manage to figure out whats popular in politcs in a real time fashion