Super Bowl 50 & the Twitterverse

What can data tell us about an event we didn't see?

Image source: https://en.wikipedia.org/wiki/Super_Bowl_50

The Event

- The **50th Super Bowl** - champions of the American Football Conference playing champions of the National Football Conference
- Audience of **111.9 million Americans** (3rd largest in US history)
- Advertising cost of **$5 million for 30 seconds**
- **Cam Newton** (**Carolina Panthers**) versus **Peyton Manning** (the **Denver Broncos**)
- And that’s all I knew up-front…

Image source: http://www.nfl.com/

The Project

- Use Natural Language Processing on a large number of Tweets
- Look at tweets using one of the main 4 hashtags (#superbowl, #superbowl50, #nfl, #sb50)
- **Can the data could tell us the key stories that happened?**

NLP – My Approach

- Extract relevant **tweets from Twitter**, pulling a large sample for each hashtag, & store
- The Twitter data would be composed of 2 sections: **Tweet text itself & the Tweeter** (who the person was, location, name, any other salient information)
- Utilise **tm text mining package** (R's most popular text mining package)
- Convert tweet content to a **corpus** (a large and structured set of texts)
- Apply **standard NLP transformations** (convert text to lowercase; remove retweets, numbers, links, spaces, URLs; remove stopwords (words of no real help - a, the, and, or, and more); stem words where needed (so that words which referenced the same thing would be treated the same))
- Build a **Document-Term Matrix** from the corpus (a matrix of the words left, to allow for analysis)
- Look at **frequency** (how often key terms are appearing/mentioned in tweets), **clustering** (do these terms fit into logical families? Can patterns be observed?), etc.
- Perform **sentiment analysis** (for each tweet, look at the positive and negative words used, and determine a sentiment score - the more negative the score, the more negative the tweet, and vice versa)
- Include **additional elements** that make sense (e.g. a **word cloud**, a **geographical analysis** of where people were tweeting from, etc.)

Getting the data – playing nice with Twitter

- Set up a Twitter app @ https://dev.twitter.com/
- Using twitteR package:
  - Create Twitter handshake

```
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

  - Extract data using searchTwitter()

```
superbowl <- searchTwitter("#superbowl", n = 10000, lang = "en", since = "2015-02-06")
superbowl50 <- searchTwitter("#superbowl50", n = 10000, lang = "en", since = "2015-02-06")
nfl <- searchTwitter("#nfl", n = 10000, lang = "en", since = "2015-02-06")
sb50 <- searchTwitter("#sb50", n = 10000, lang = "en", since = "2015-02-06")
```

  - searchTwitter() – text of the tweet; screenname of Tweeter; when tweet was created; was the status favourited (and how many times); longitude/latitude of user; and more…
  - Limitation: Twitter only returns subset of tweets from the last week, and biased towards recency (so all tweets are from 1-2 days after the event)

“I need to make a corpse?” – The Corpus

- Strip out all retweets using strip_retweets()

```
superbowl_no_rt <- strip_retweets(superbowl)
```

- Store data in data frames

```
superbowl_df <- twListToDF(superbowl_no_rt)
```

- Create a **corpus** – a large collection of documents

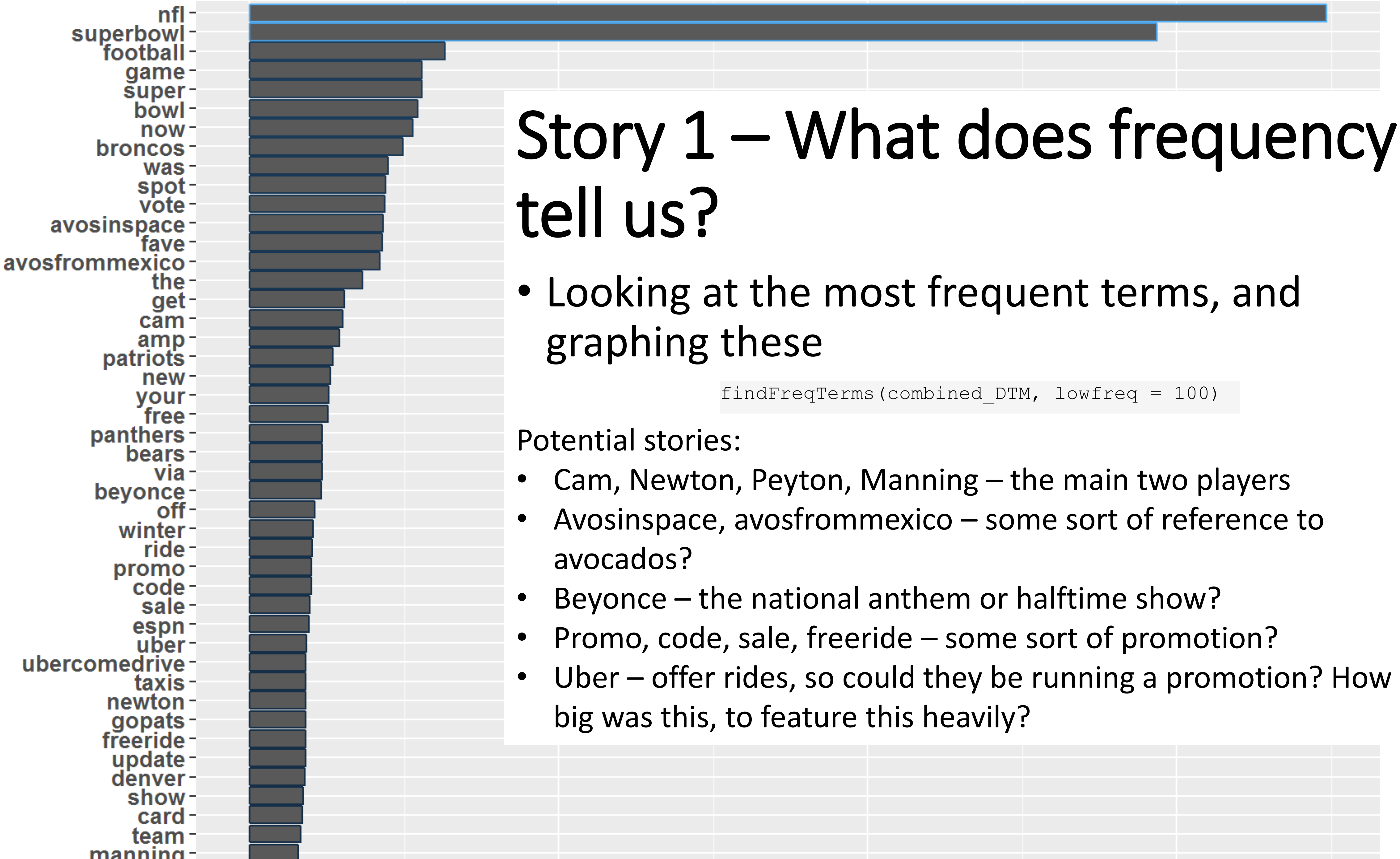
```
combined_corpus <-
Corpus(VectorSource(combined_df$text))
```

- Perform a number of standard transformations – removing punctuation, whitespace, URLs, stopwords (“and”, “the”, etc.); make the corpus lowercase

```
combined_corpus <- tm_map(combined_corpus,
removePunctuation)
```

- Create a Document Term Matrix (matrix that describes the frequency of terms that occur in a collection of documents) and a sparse DTM (ignore terms that with frequency lower than a given threshold, making our remaining terms more relevant)

```
combined_DTM <-
DocumentTermMatrix(combined_corpus)

combined_DTMs <-
removeSparseTerms(combined_DTM, 0.99)
```



Story 1 – What does frequency tell us?

- Looking at the most frequent terms, and graphing these

```
findFreqTerms(combined_DTM, lowfreq = 100)
```

Potential stories:

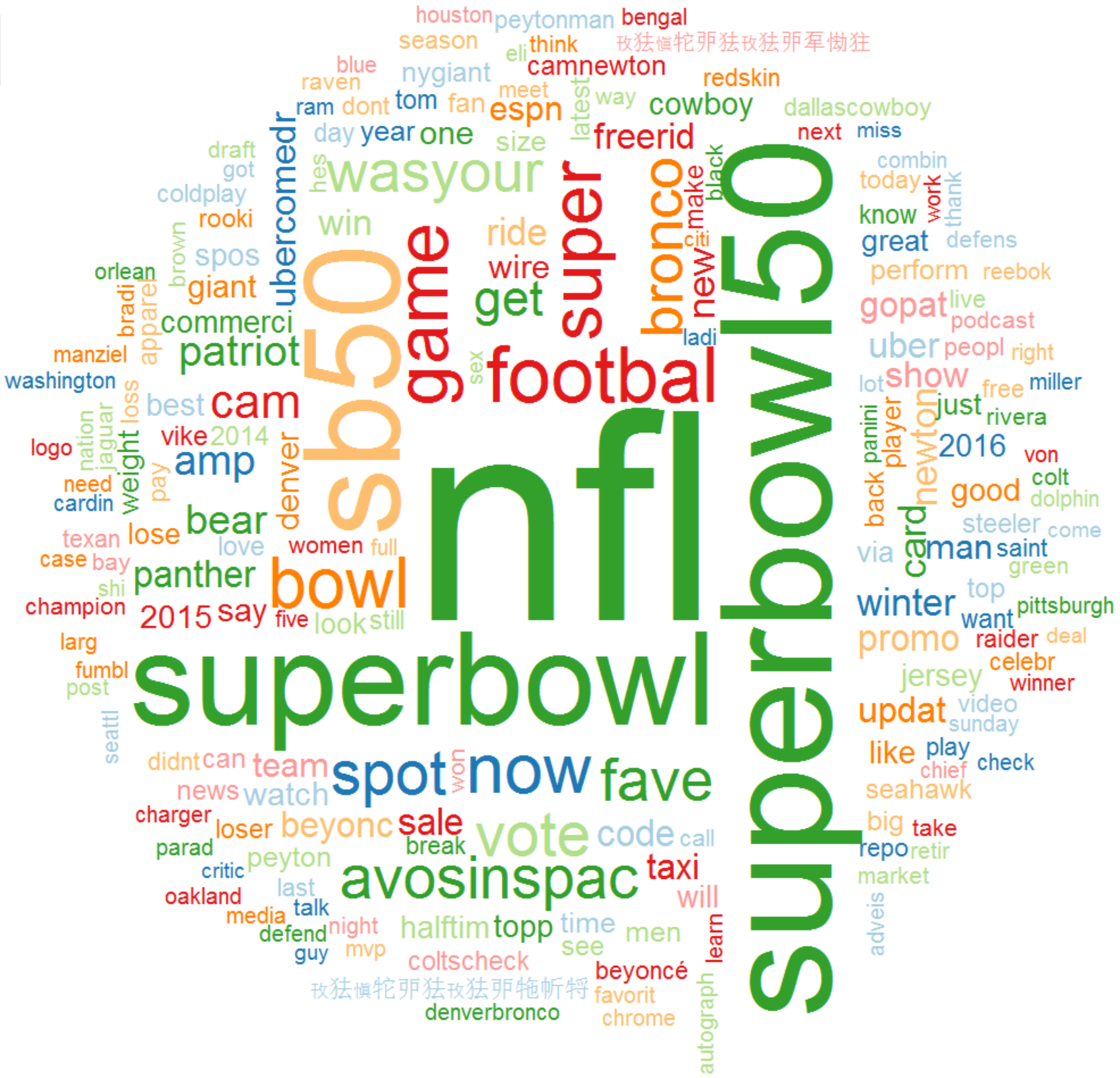
- Cam, Newton, Peyton, Manning – the main two players
- Avosinspace, avosfrommexico – some sort of reference to avocados?
- Beyonce – the national anthem or halftime show?
- Promo, code, sale, freeride – some sort of promotion?
- Uber – offer rides, so could they be running a promotion? How big was this, to feature this heavily?

Frequency reimagined

- A wordcloud allows us to see the term frequency in a more visual way

```
wordcloud(combined_text_corpus, min.freq =
min_freq, scale = c(8, 0.5), rot.per=.15, colors
= brewer.pal(8, "Paired"),  random.color = TRUE,
random.order = FALSE, max.words = Inf)
```

- The hashtags dominate proceedings, but we can see some of other terms really jump out – "camnewton", "avosinspace", "beyonce", "uber". We'll look at these in more detail, next
- There are a number of other similar terms – teams ("seahawk", "raider", "dallascowboy"), sports ("reebok", "apparel"), television channels/networks ("espn")
- Like in anything that exists on the Internet, "sex" is in there. I have no idea how, but this is the Internet, I guess



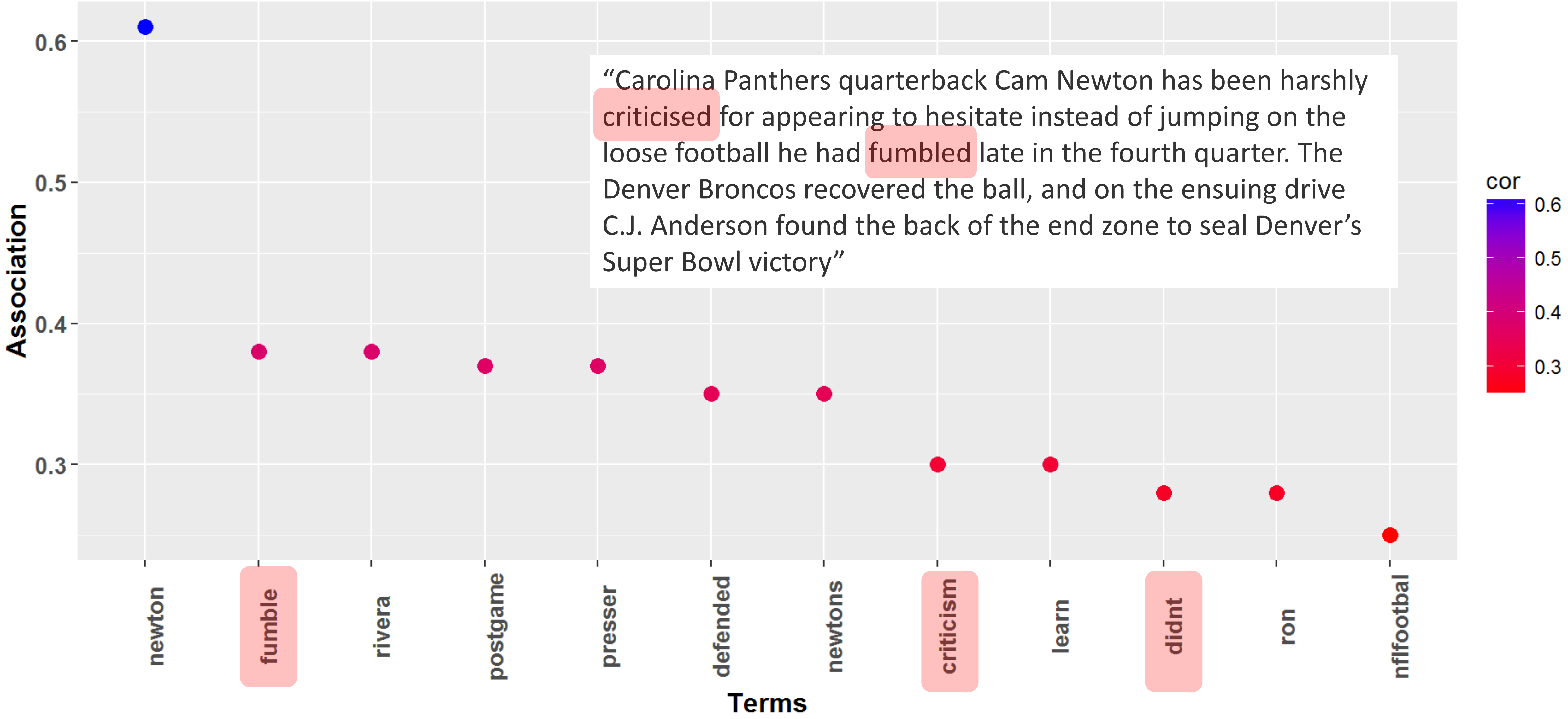
Story 2: A level deeper - word associations

- For some of the most frequent terms from our analysis and word cloud, let’s look at what words they are most frequently associated with

```
uber_assoc <- findAssocs(combined_DTM, "uber", assoc)
```

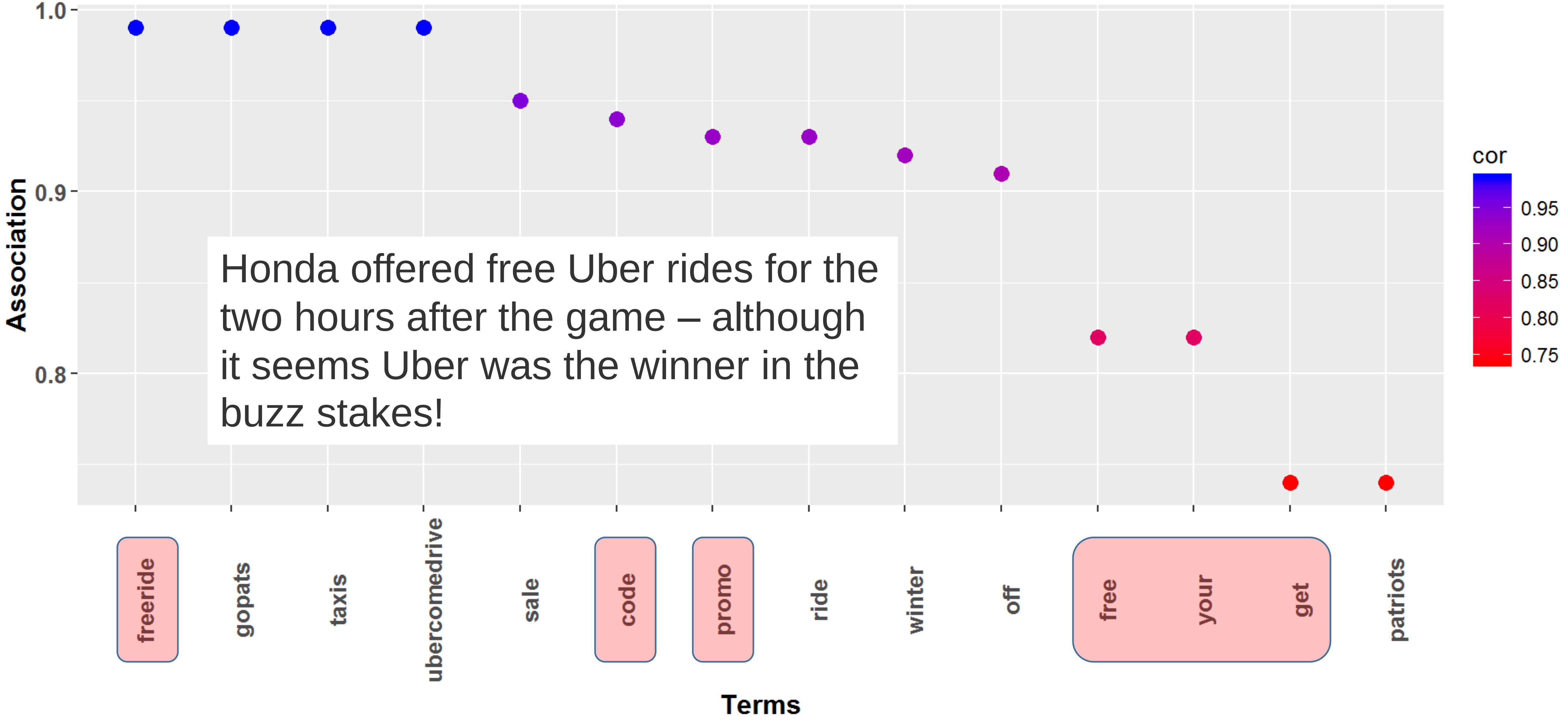
- The accompanying report has a more extensive list – we’ll look at the most interesting findings

Cam Newton – what happened?



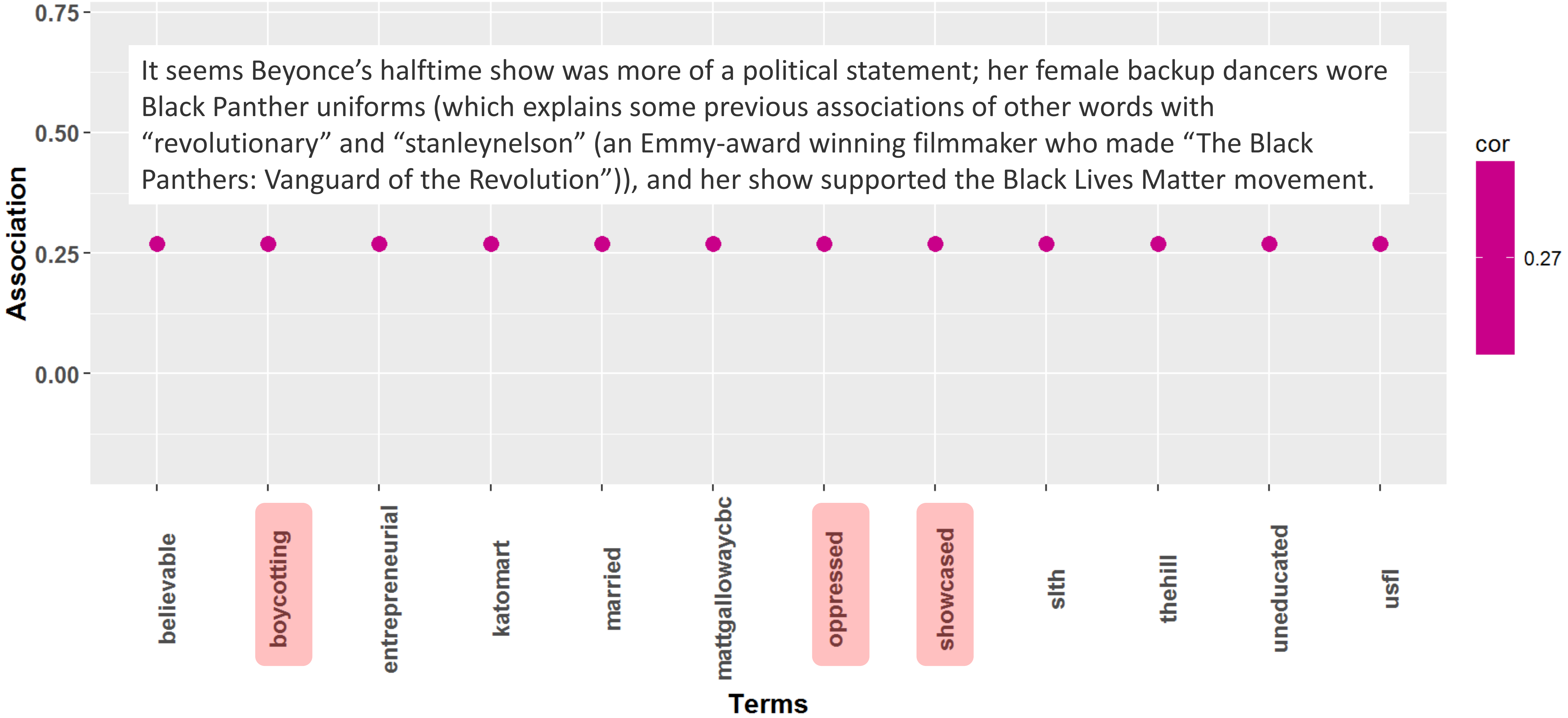
"Carolina Panthers quarterback Cam Newton has been harshly criticised for appearing to hesitate instead of jumping on the loose football he had fumbled late in the fourth quarter. The Denver Broncos recovered the ball, and on the ensuing drive C.J. Anderson found the back of the end zone to seal Denver's Super Bowl victory"

Uber (& Honda) – frenemies?



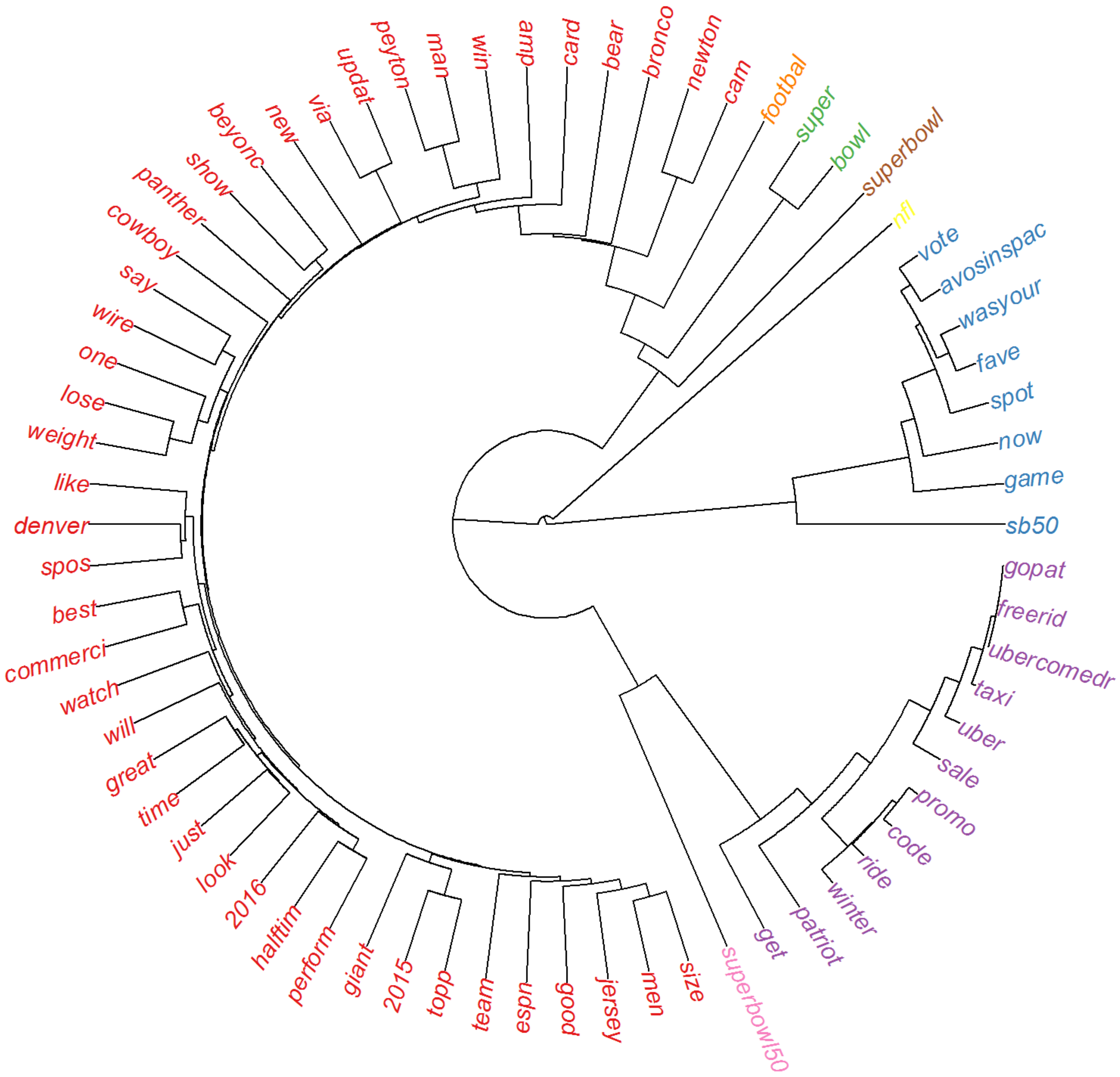
Beyonce – a political stance?

It seems Beyonce’s halftime show was more of a political statement; her female backup dancers wore Black Panther uniforms (which explains some previous associations of other words with “revolutionary” and “stanleynelson” (an Emmy-award winning filmmaker who made “The Black Panthers: Vanguard of the Revolution”)), and her show supported the Black Lives Matter movement.



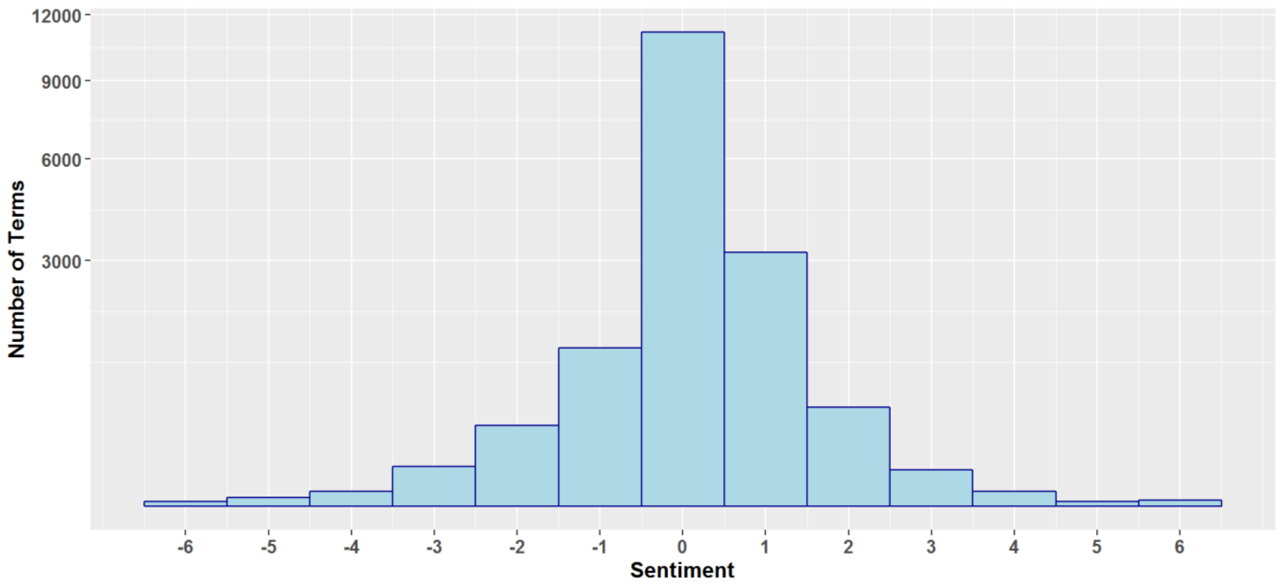
Cluster's Last Stand

- Clustering terms allows us to see what is/could be related
- Rather than a regular dendogram, we'll use a package (Ape) to create a more visually-appealing dendogram
- We can see a number of distinct relationships:
  - The largest is **the game itself** - teams, half-time show and performers, key players, etc.
  - An **Uber** cluster pulls together all the terms related to the free ride promo. With no mention of Honda, it seems Uber won the buzz battle
  - An **Avo's** cluster, which seems to be some sort of spot asking people to vode for their favourite...something? Looking into this, a company called **Avocados From Mexico** ran a commercial, pusing the "avosinspace" hashtag (hence the frequency of the term), and a number of companies are now asking people to vote for their favourite Superbowl commercial (and it seems this was it).



Story 3: Tell me how you feel…

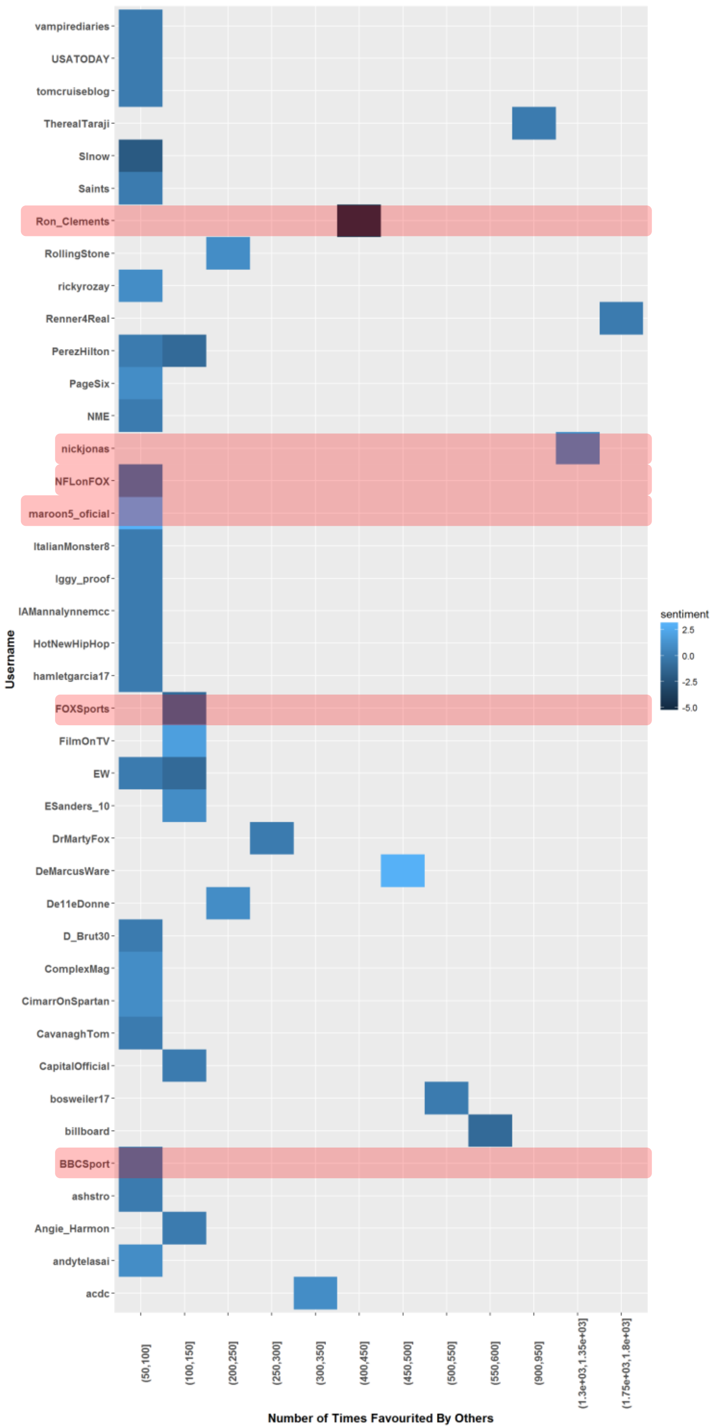
- Let's look at the sentiment of the tweets – were people positive? Negative? Neutral? Angry? Sad? Happy?
- We'll use a basic sentiment analysis model for this:
    - Take in a piece of text
    - Reviews it for positive and negative keywords,
    - Scores +1 for positive words found, -1 for negative words found
    - Tally the scores, giving a sentiment score for each piece of text (in our case, each tweet).
- Positive and negative word lists are from Hu and Liu's Opinion Lexicon (https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon)
- We can see that the vast majority of tweets are sentiment neutral (0), then 1 (positive), -1 (negative) and so forth. Overall, we can sentiment is hugely neutral, with the slightest bias towards positive; this is borne out when we look at the mean and median values



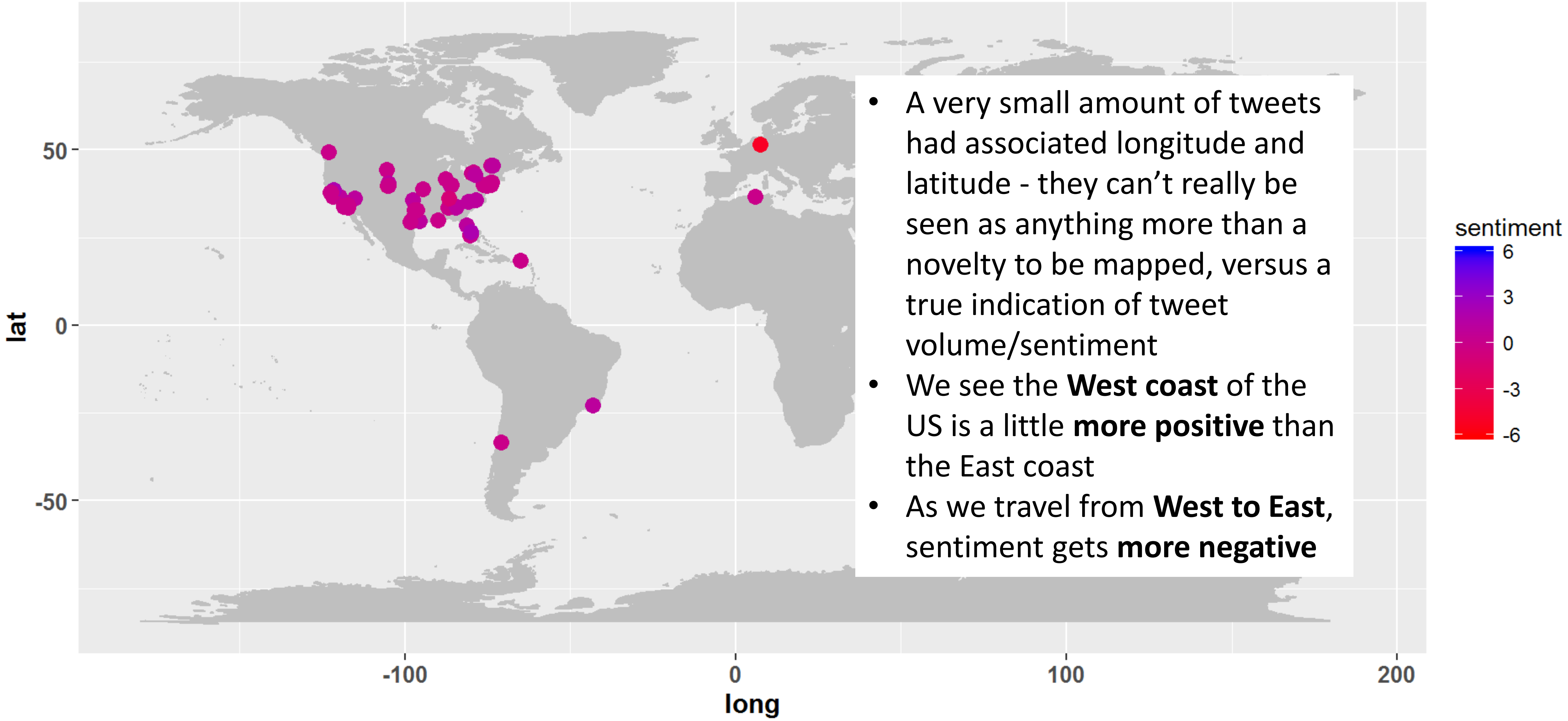

```
mean(combined_df$sentiment)
[1] 0.1347915

median(combined_df$sentiment
)
[1] 0
```

Story 4: Feel the heat(map)

- Let's look at the sentiment of the most popular tweets (those tweets favourited more than 50 times), and who tweeted these
- We create a **bucket of the favouriteCount variable** - since this is a continuous variable, we want to reign it in, so we'll look at buckets bracketed by 50 (0 - 50 favourites, 51 - 100, etc.)
- Limitation: we should expect the heatmap to have a number of blank spaces, as it doesn't have data points for every bucket
- We can see that the **most-favourited tweets** were primarily **neutral to negative**; that those from the **sports networks** were **more neutral** (BBC, FOX Sports); a couple of **celebrities** got in on the act with **positive** tweets (Maroon 5, Nick Jonas); and **Ron Celements** (an NFL reporter) had the **most-favourited negative tweet**.



Story 5: We know where you live



- A very small amount of tweets had associated longitude and latitude - they can’t really be seen as anything more than a novelty to be mapped, versus a true indication of tweet volume/sentiment
- We see the **West coast** of the US is a little **more positive** than the East coast
- As we travel from **West to East**, sentiment gets **more negative**

Summary

- From Beyonce's political stance, to Cam Newton's fumble, to Avocado's in Space, to Honda and Uber's mega-deal which everyone was talking about, it seems that a set of tweets can help us see stories in the data

- For more information:
  - Full R code, R Markdown report @ https://github.com/ivanheneghan/
  - Blog post @ http://www.prettypicturestellstories.com/