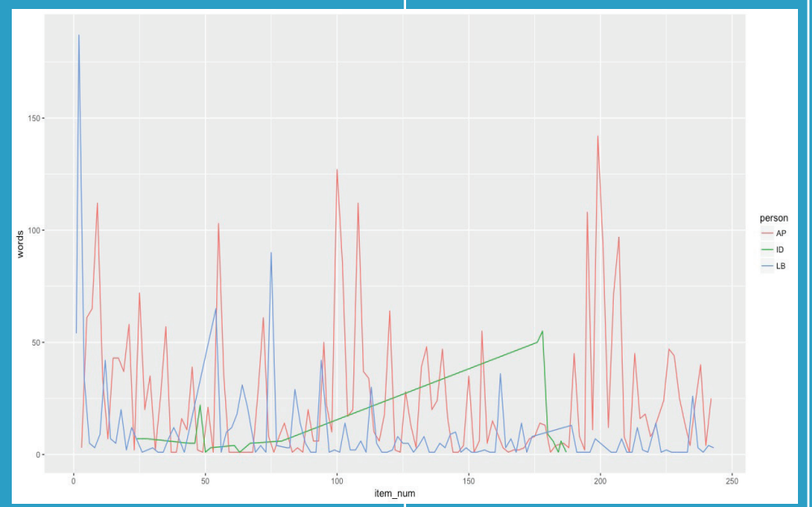
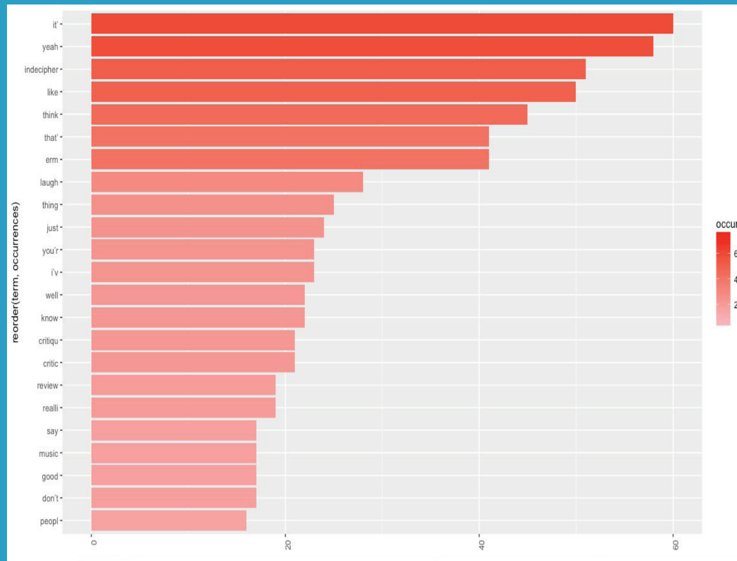
REMIXING the REMIX and 'STEMS'

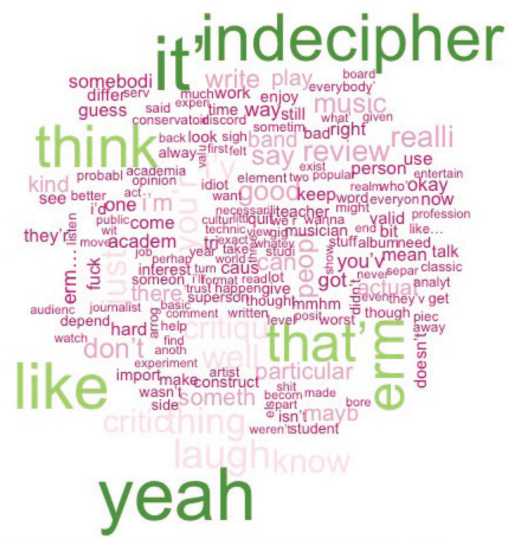
CRAIG HAMILTON

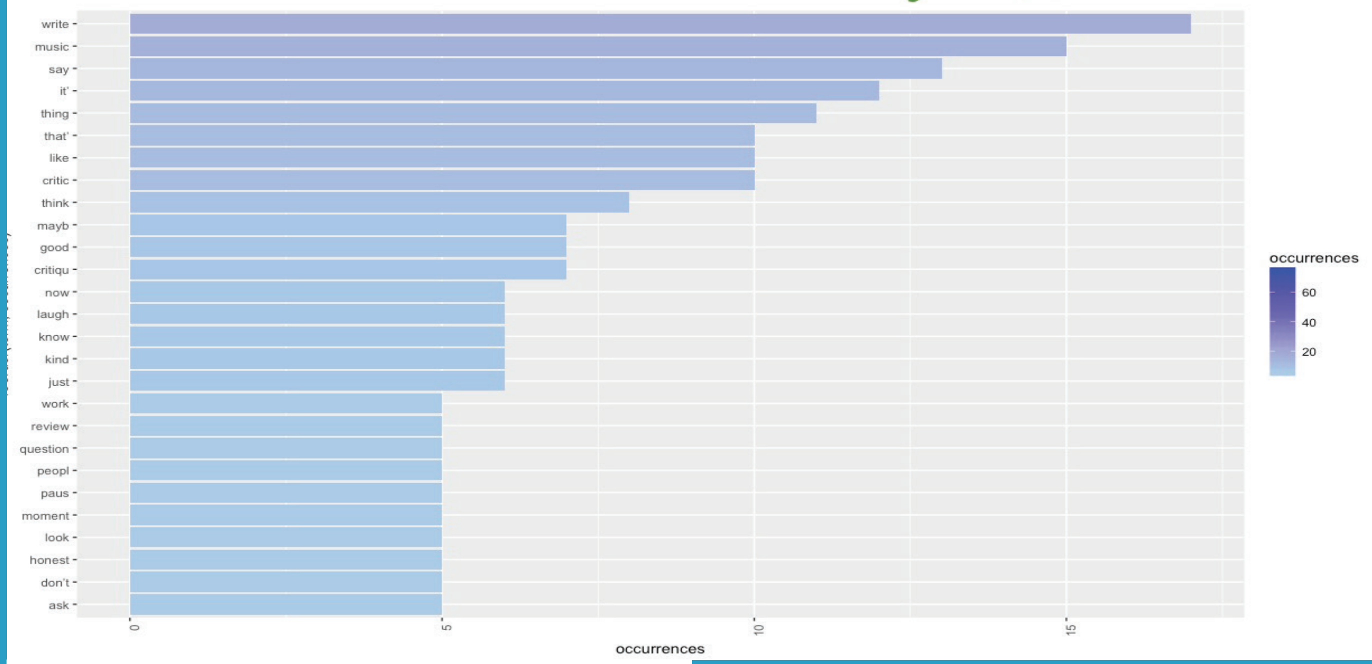
```
#NOTES FOR READERS
#This experimental writing piece for Riffs Issue 3 is presented as a coding script.
#Lines starting with a hashtag symbol (#) mean code on those lines is not run. These lines are usually used to insert instructions and context into coding scripts so that people using the script can understand the steps within. The purpose of this piece is to explore the following question:
# A machine will not question whether it is right or wrong to hate a rock critic (unless explicitly 'told' to do so), but can machine processing (such as that used in this script) help us think about that question?

#The data set and this script has been provided at the following link:
 https://www.dropbox.com/sh/yi8c536uoz1jgqi/AAAHx1rZTzeCuXie9nw-66sAka?dl=0

#If you have an installation of the R Software on your machine, you can replicate this work.
```

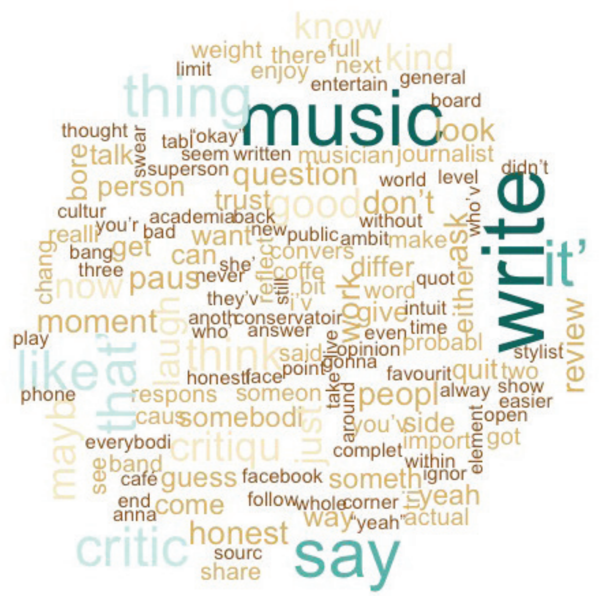
```
#STEP 1: READING IN THE DATA

#The transcript of the original interview between Lyle Bignon and Anna Palmer, along with the remix provided by the Riffs team, has been inserted into a dataframe. This can be read into the R environment as an object.

Sys.setlocale('LC_ALL','C')
## - see http://r.789695.n4.nabble.com/Strings-from-different-locale-td3023176.html

library(xlsx)
interview <- read.xlsx('riffs.xlsx', 1, stringsAsFactors = F)
dim(interview) #the database has 259 observations (rows), and 4 variables (columns)

names(interview)
 #the 4 variables are:item_num (the sequential number of each item); person (initials identifying the speaker/writer); text (the words said/written); int_write (whether the text is from the interview (int) or the write up (write))

unique(interview$person)
#will tell us that the 11 individuals speaking/writing in the dataframe are: LB: the interviewer; AP: the interviewee; ID: the photographer; DK / NG / SS / SR / AD / MG / IT: the Riffs writers

#To create some simple visualisations we can first must count the words and characters in each entry:

library(magrittr)
library(dplyr)
interview$chars <- sapply(interview$text, function(x) nchar(x))
interview$words <- sapply(strsplit (interview$text, "\\s+"), length)

#In order to perform some exploratory automated textual analysis, we need to prepare the data by first removing all punctuation, whitespace, and by stemming all words to their roots removing commonly occurring 'stopwords', and then creating a Document Term Matrix
library(tm)
library(stringr)
story_stem <- str_replace_all(interview$text, "@", "")
story_stem <- str_replace_all(story_stem, "@\\w+", "")
story_stem <- stemDocument(story_stem)
story_stem <- removePunctuation(story_stem)
story_stem <- tolower(story_stem)
story_stem <- stripWhitespace(story_stem)
interview <- cbind(interview, story_stem)
dtm.control <- list(
 tolower =T,
 removePunctuation =T,
 removeNumbers =T,
 stopwords = c(stopwords("english")),
 stemming =T,
 wordLengths = c(3,Inf),
 weighting = weightTf
)

#We can now split the Interview data frame into two parts:

transcript <- interview %>%
  filter(int_write == "int")

 # = only the entries from the transcript

writeup <- interview %>%
 filter(int_write == "write")
 # = and only those from the write up

library(ggplot2)

#2: FIRST QUESTION:

#Looking firstly at the interview transcript, we may want to look at who talked, and when by visualising those word counts across the interview

transcript %>%
  ggplot() +
   aes(x = item_num, y = words, colour = person) +
geom_line()
#We can see from the visualisation above that the interviewer (blue line) talked a lot more at the beginning of the interview, but eventually the interviewee (red line) began to talk more.

A brief lull in conversation between the two can be see by the interjection of the photographer (green). This is perhaps what we may expect to see from an interview process. The interviewer sets the scene, gets the conversation going, before eventually settling back and letting their subject talk.

#We may also want to look at the words each person used. We can do this in the first instance with some basic counts. We first create document term matrices for both the transcript and write up elements.

dtm_transcript <- DocumentTermMatrix(Corpus(VectorSource(transcript$story_stem)),

control = dtm.control)
dtm_transcript <- removeSparseTerms(dtm_transcript,0.999)
dim(dtm_transcript)

#this creates a DTM of the transcript that has 243 entries, and 681 terms

matrix <- as.matrix(dtm_transcript)

#to get the frequency of occurrence of each word in the corpus, we simply sum over all rows to give column sums:

freq <- colSums(as.matrix-(dtm_transcript))

#and then visualise these in descending order

wf=data.frame(term=names(-freq),occurrences=freq)
ggplot(subset(wf, freq>15), aes(x = reorder(term, occurrences), y = occurrences, fill = occurrences)) +
geom_bar(stat="identity") +
 theme(axis.text.x=element_text(angle=90, hjust=1)) +
   coord_flip(xlim = NULL, ylim = NULL, expand = TRUE) +
  scale_fill_gradient2(low = "white", mid = "pink", high = "red", limits = c(5, 75))

#We can also create a wordcloud of what was said between interview and interviewee.

library(wordcloud)
set.seed(42)
wordcloud(names(freq),freq,min.freq=3,colors=brewer.pal(6,"PiYG"))
```



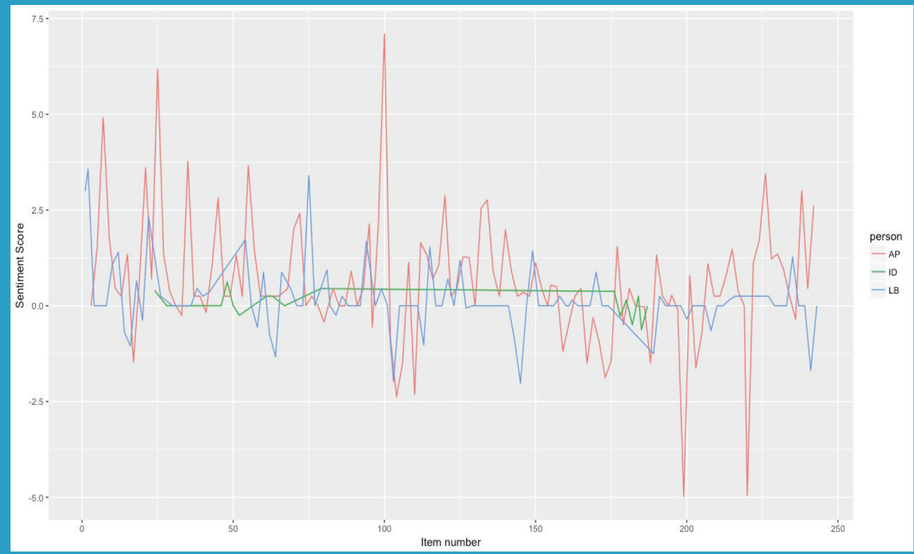


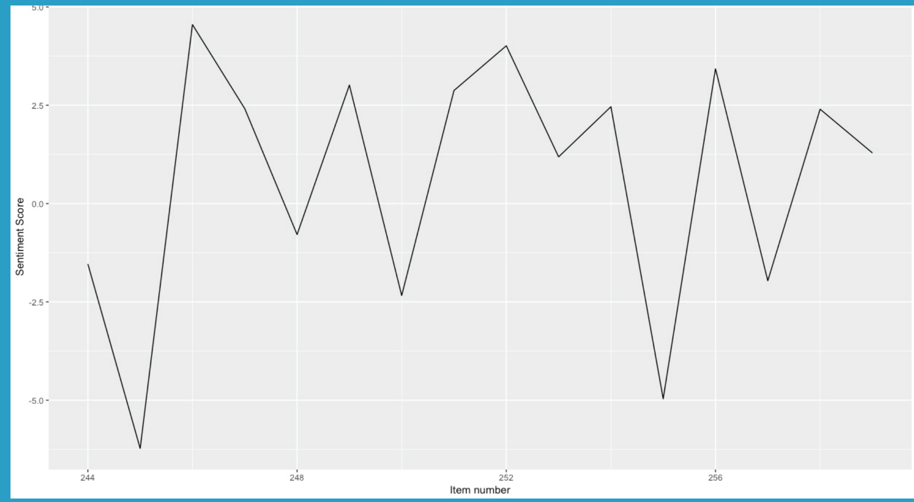







```
#Finally, we can run 'Sentiment
Analysis' to get some idea of the emo-
tional valence of the conversation
during the interview, and during the
write up.

###SENTIMENT ANALYSIS
library(syuzhet)
library(scales)
library(reshape2)
library(dplyr)

mySentiment <- get_nrc_sentiment(in-
terview$text)

head(mySentiment, 5)

interview <- cbind(interview, mySen-
timent)

syuzhet_sent <- get_sentiment(inter-
view$text, method = "syuzhet")

interview <- cbind(inter-
view, syuzhet_sent)

bing_sent <- get_sen-
timent(interview$text,
method = "bing")

interview <- cbind(inter-
view, bing_sent)

afinn_sent <- get_sen-
timent(interview$text,
method = "afinn")

interview <- cbind(inter-
view, afinn_sent)

nrc_sent <- get_sen-
timent(interview$text,
method = "nrc")

interview <- cbind(inter-
view, nrc_sent)

sent_scores <- c(syuzhet_
sent + bing_sent + afinn_
sent + nrc_sent)

interview <- mutate(in-
terview, sent_score_ave =
sent_scores/4)

interview <- mutate(in-
terview, sent_by_word =
sent_score_ave/words)

#By visualising these
results….

interview %>%
     filter(int_write ==
"int") %>%
  ggplot() +
     aes(item_num, sent_
score_ave, colour = per-
son) +
   geom_line() +
     xlab("Item number") +
ylab("Sentiment Score")


interview %>%
     filter(int_write ==
"write") %>%
  ggplot() +
     aes(item_num, sent_
score_ave) +
```






```
  geom_line() +
   xlab("Item number") +

#...We can see that –
according to the combined
scores of a number of
different Sentiment
Analysis algorithms, at
least – the 'mood' of the
conversation between
interviewer and
interviewee fell during
the course of the process.

Interestingly, the
Write-Club write up
tended to follow and pick
up on these ups and downs.
The 'highs' of the
early part of the
interview, then the fall
during the second half,
and finally the positive
note struck at the end,
are all reflected in these
numbers.
```

Craig Hamilton is a Research Fellow in the School of Media at Birmingham City University.

His research explores contemporary popular music reception practices and the role of digital, data and Internet technologies on the business and cultural environments of music consumption.

This research is built around the development of The Harkive Project (www.harkive.org), an online, crowd-sourced method of generating data from music consumers about their everyday relationships with music and technology. Craig is the co-Managing Editor of Riffs: Experimental Research on Popular Music.