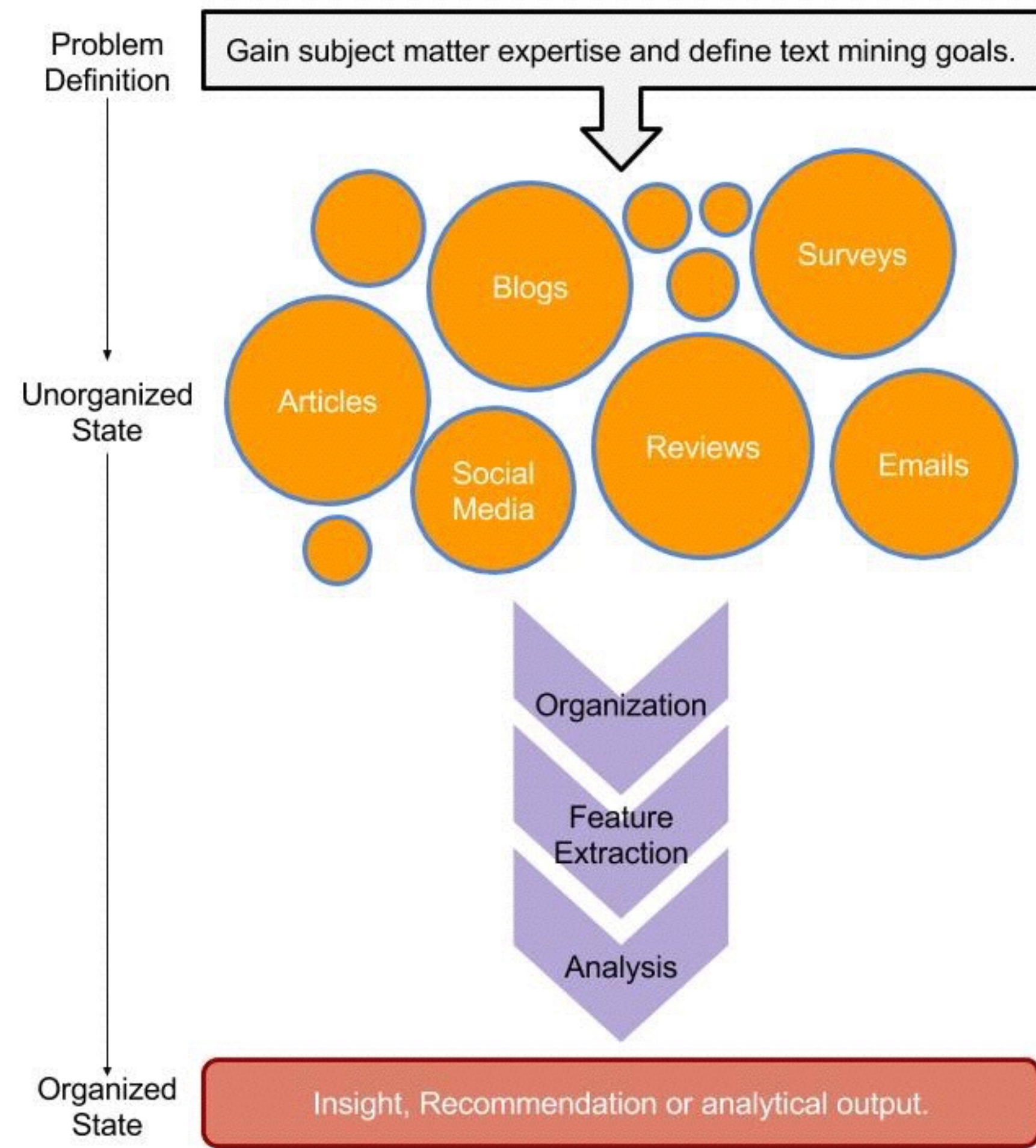




INTRO TO TEXT MINING: BAG OF WORDS

Amazon vs. Google

Remember the workflow?



1 - Problem definition & specific goals

2 - Identify text to be collected

3 - Text organization

4 - Feature extraction

5 - Analysis

6 - Reach an insight,
recommendation or output

A case study in HR analytics

1. Problem definition

Which company has better work life balance? Which has better perceived pay according to online reviews?

2. Unorganized state



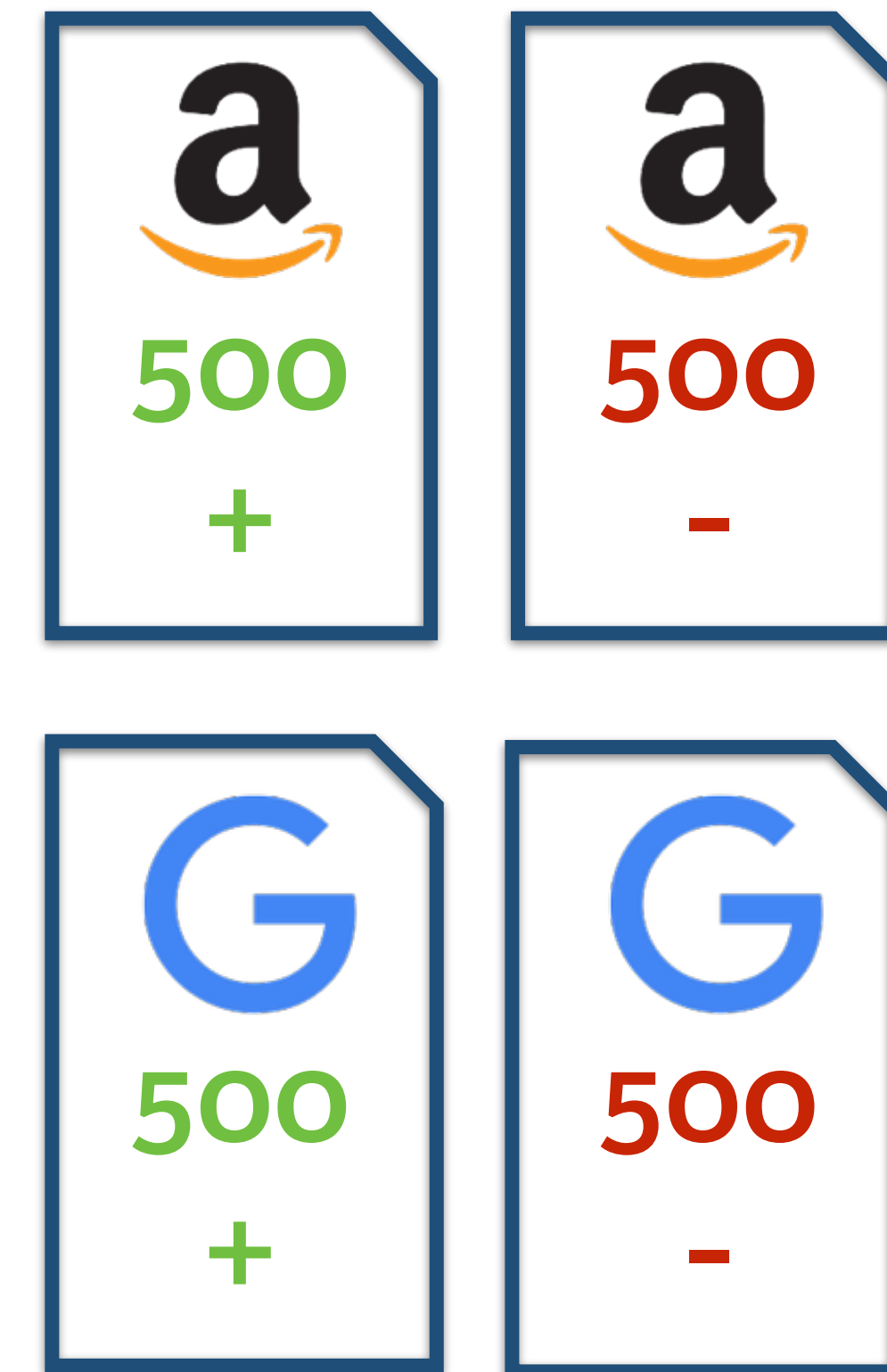
3. Organization

4. Feature Extraction

5. Analysis

6. Organized state

Insight, recommendation, analytical output





INTRO TO TEXT MINING: BAG OF WORDS

Let's practice!



INTRO TO TEXT MINING: BAG OF WORDS

Text organization

Text organization with qdap

```
# qdap cleaning function
qdap_clean <- function(x) {
  x <- replace_abbreviation(x)
  x <- replace_contraction(x)
  x <- replace_number(x)
  x <- replace_ordinal(x)
  x <- replace_symbol(x)
  x <- tolower(x)
  return(x)
}
```


Text organization with tm

```
# tm cleaning function
tm_clean <- function(corpus) {
  tm_clean <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeWords,
                   c(stopwords("en"), "Google", "Amazon", "company"))
  return(corpus)
}
```

Cleaning your corpora





INTRO TO TEXT MINING: BAG OF WORDS

Let's practice!



INTRO TO TEXT MINING: BAG OF WORDS

Feature extraction and analysis

Feature extraction

```
> # Create bigram TDM
> amzn_p_tdm <- TermDocumentMatrix(
  amzn_pros_corp,
  control = list(tokenize = tokenizer)
)
```



	Review 1	Review 2	...	Review N
Bigram 1	0	0	0	0
Bigram 2	1	1	0	0
Bigram 3	1	0	0	0
...	0	0	1	1
Bigram M	0	0	1	0

Term Document Matrix (TDM)

Get term frequencies

```
> # Convert TDM to matrix
> amzn_p_m <- as.matrix(amzn_p_tdm)

> # Compute term frequencies
> amzn_p_freq <- rowSums(amzn_p_m)

> # Sort in decreasing order of frequency
> term_frequency <- sort(amzn_p_freq, decreasing = TRUE)

> View the top 5 most frequent bigrams
> term_frequency[1:5]
```

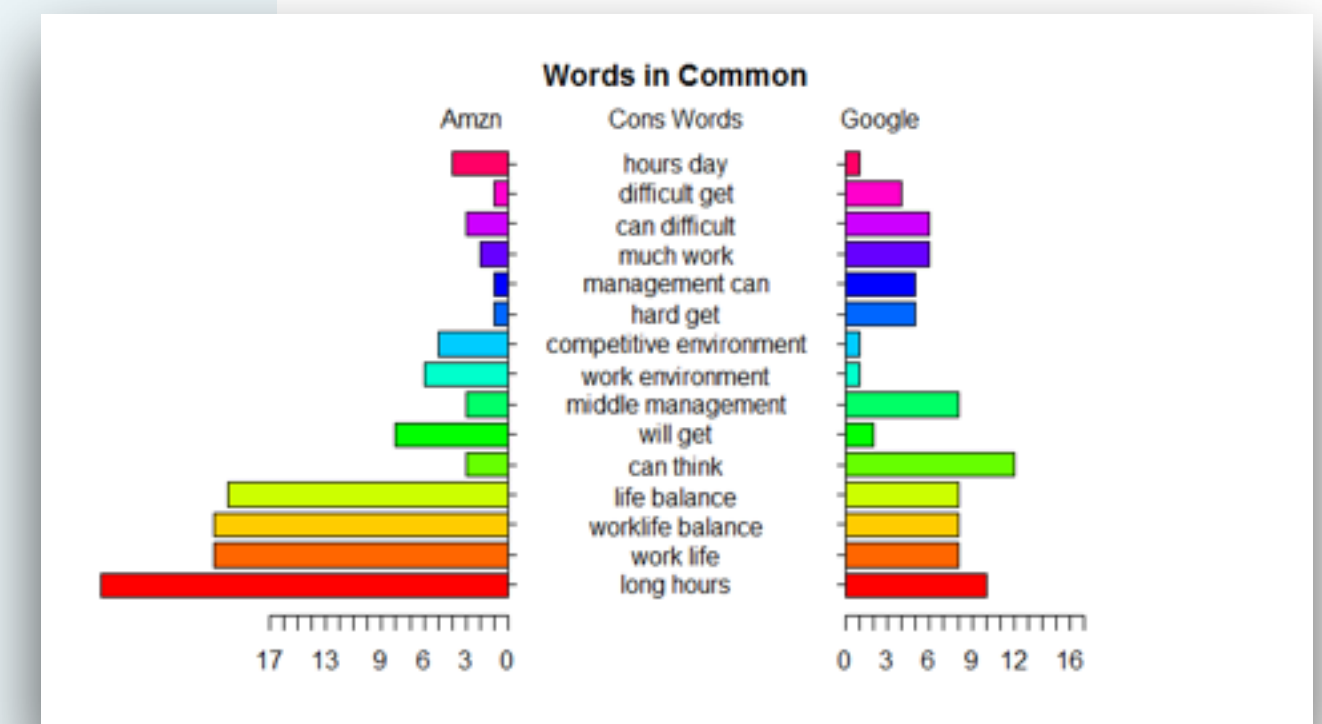
good pay	great benefits	smart people
25	24	20
place work	fast paced	
17	16	

Create visuals with plotrix

```
> # Find common words
> common_words <- subset(all_tdm_m,
                          all_tdm_m[, 1] > 0 & all_tdm_m[, 2] > 0)
> difference <- abs(common_words[, 1] - common_words[, 2])
> common_words <- cbind(common_words, difference)
> common_words <- common_words[order(common_words[, 3],
                                      decreasing = TRUE), ]

> # Create data frame: top 15 words
> top15_df <- data.frame(x = common_words[1:15, 1],
                        y = common_words[1:15, 2],
                        labels = rownames(common_words[1:15, ]))

> # Make pyramid plot
> pyramid.plot(top15_df$x, top15_df$y, labels = top15_df$labels,
              gap = 12, main = "Words in Common", unit = NULL,
              top.labels = c("Amzn", "Cons Words", "Google"))
```





INTRO TO TEXT MINING: BAG OF WORDS

Let's practice!



INTRO TO TEXT MINING: BAG OF WORDS

Reach a conclusion

Time to reach a conclusion!

Problem definition

Which company has better work life balance? Which has better perceived pay according to online reviews?

Unorganized state



Organization

Feature Extraction

Analysis

Organized state

Insight, recommendation, analytical output



INTRO TO TEXT MINING: BAG OF WORDS

Let's practice!

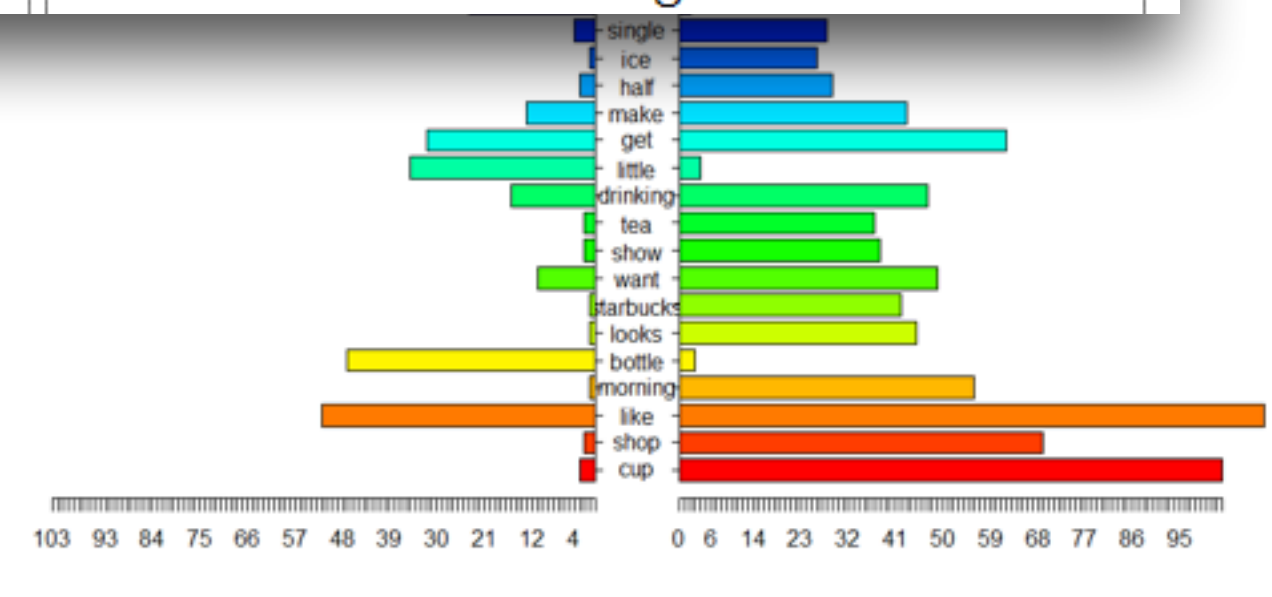
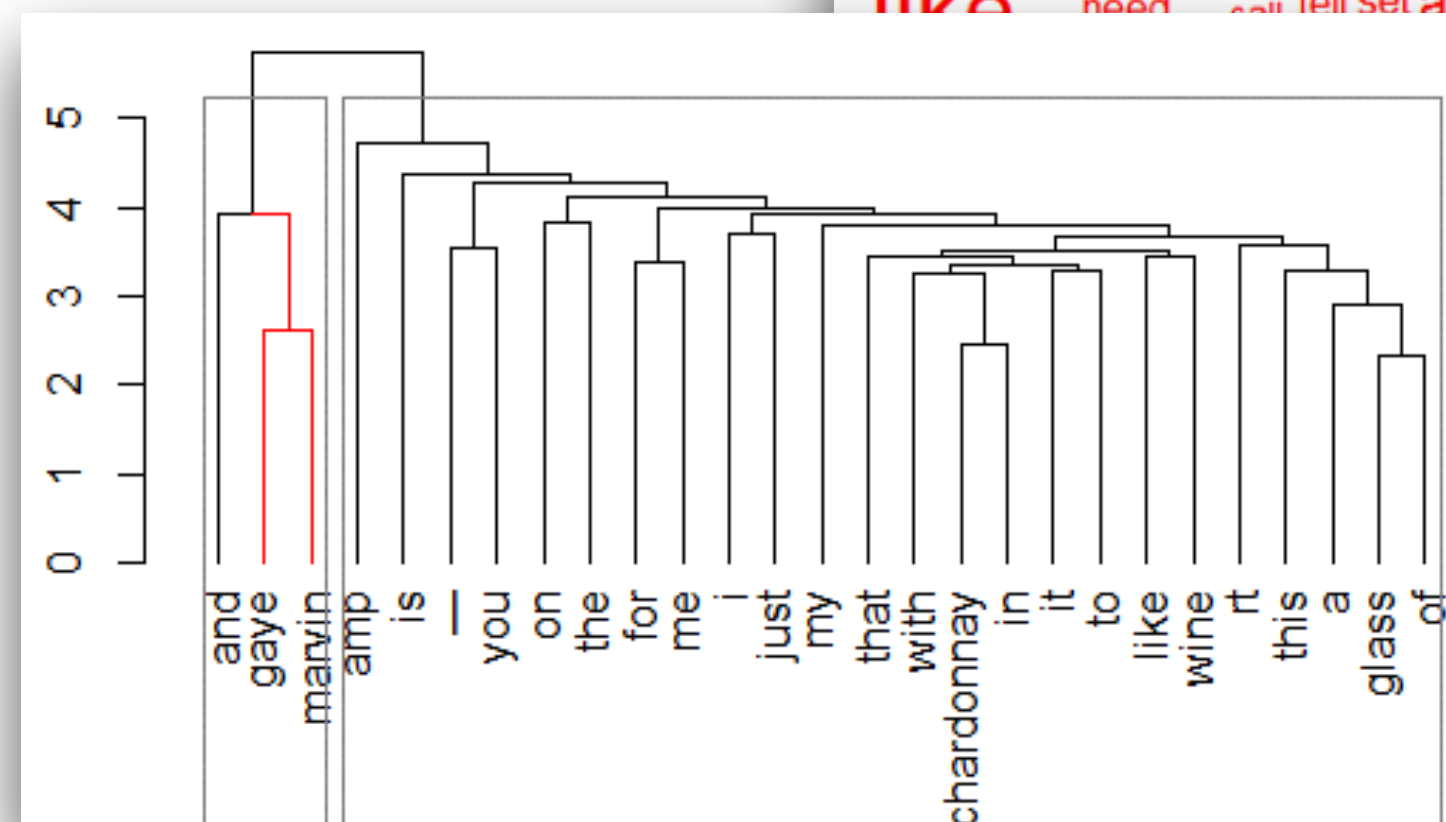


INTRO TO TEXT MINING: BAG OF WORDS

Congratulations!

In this course, you learned how to...

- Organize and clean text data
- Tokenize into unigrams & bigrams
- Build TDMs & DTMs
- Extract features
 - Top terms
 - Word associations
- Visualize text data





INTRO TO TEXT MINING: BAG OF WORDS

Get to work!