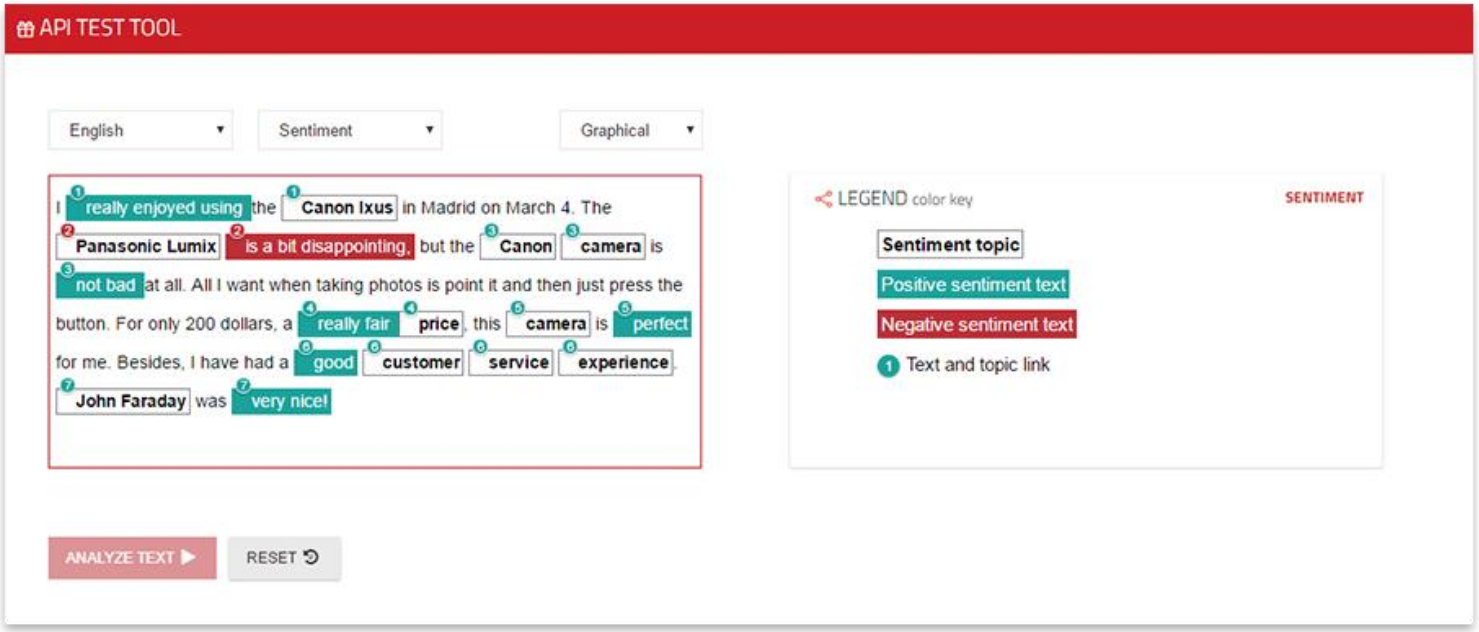
Sentiment analysis

SIMPLE TEXT ANALYTICS WITH R AND TIDYTEXT

PEETER TINITS 07.02.2018

Sentiment analysis of storylines

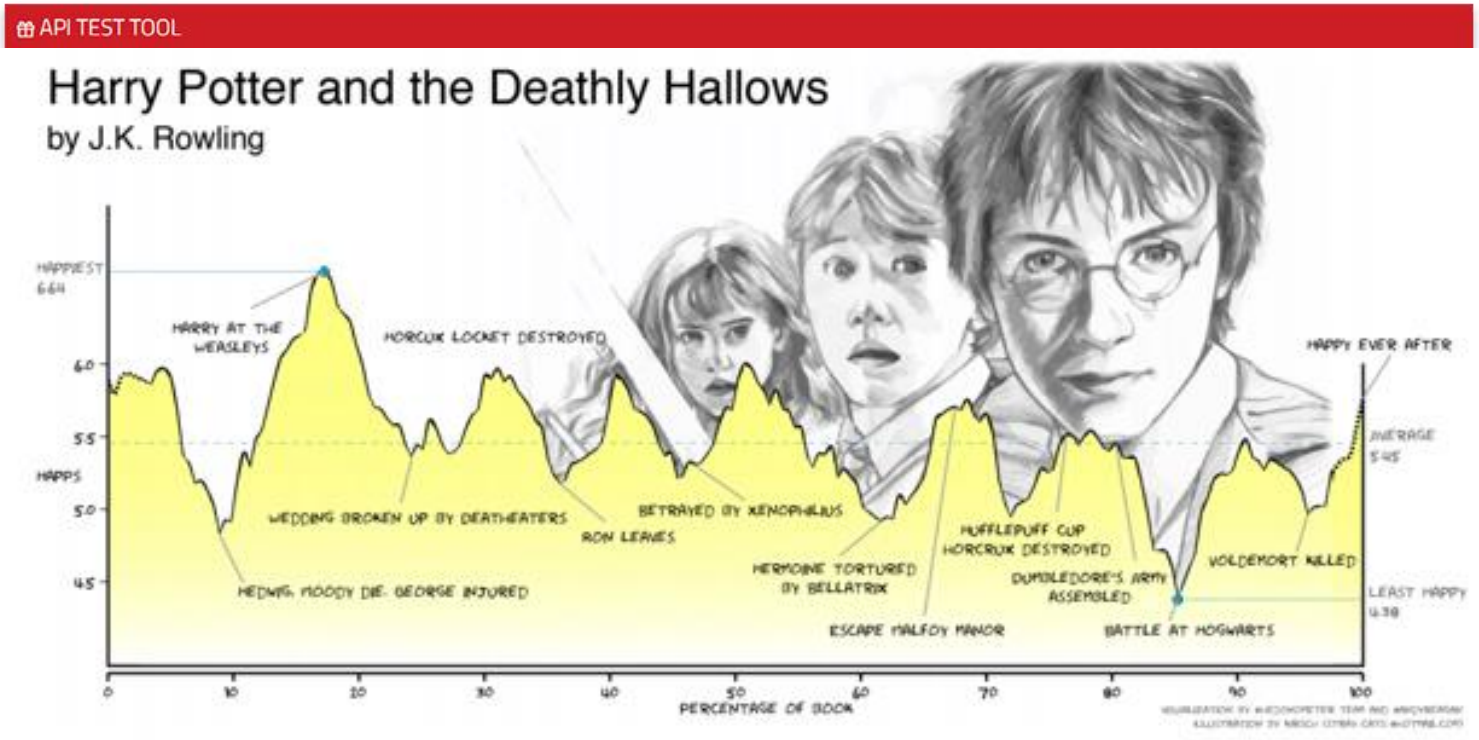
Words and phrases with known sentiment can be picked out in texts



Bitext tool

Sentiment analysis of storylines

Words and phrases with known sentiment can be picked out in texts



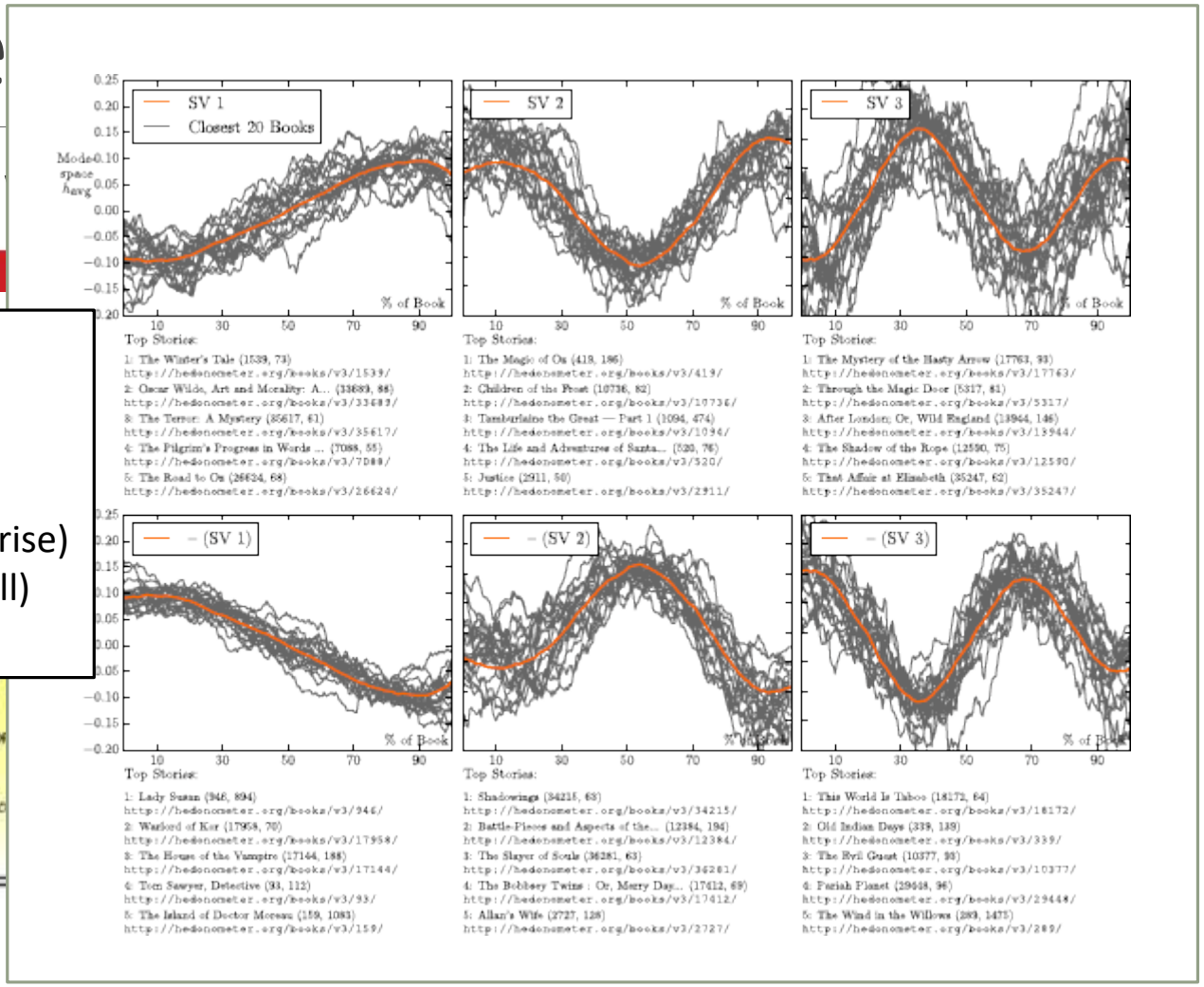
[The story arc of Harry Potter according to AI (Hedonometer team and Andrew Reagan)]

Sentiment analysis of storyline

Words and phrases

API TEST TOOL

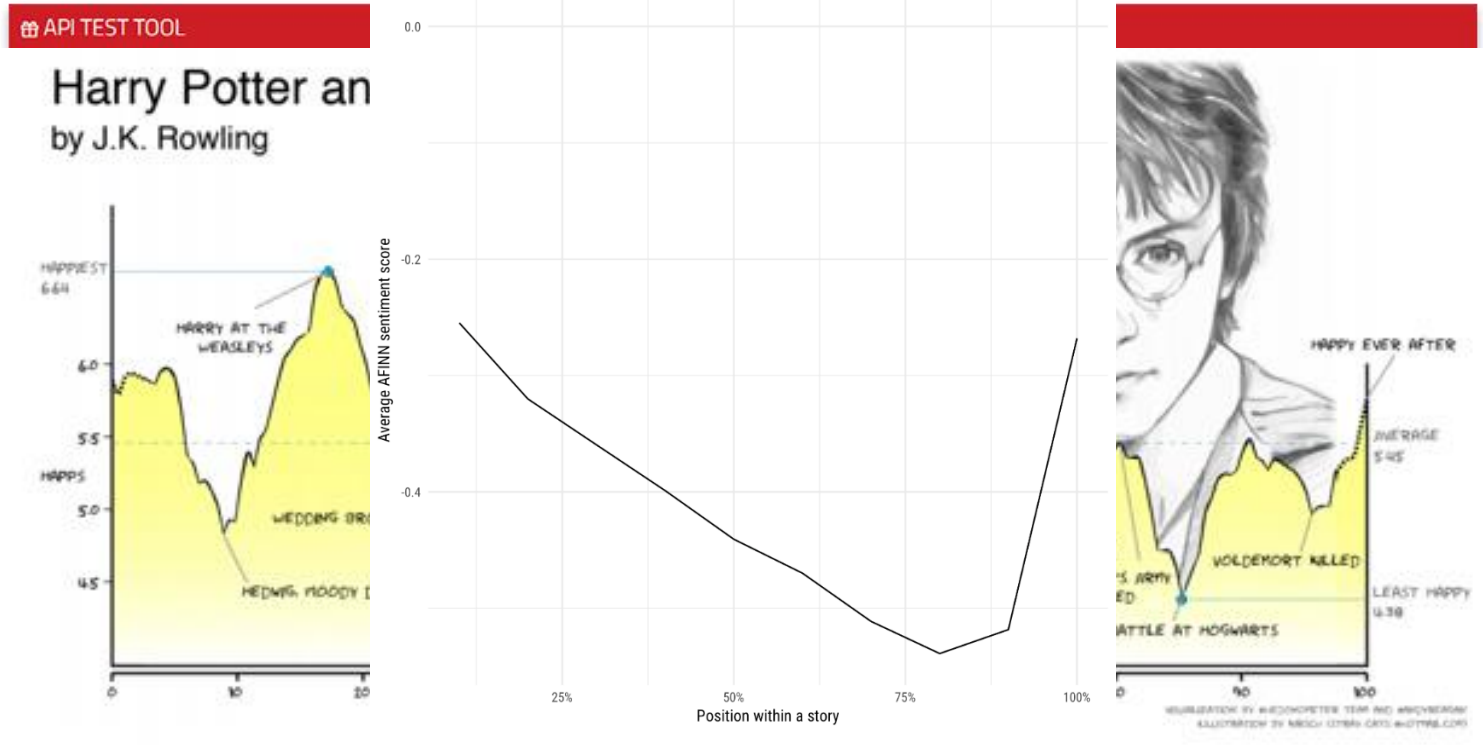
1. Rags to Riches (rise)
2. Riches to Rags (fall)
3. Man in a Hole (fall then rise)
4. Icarus (rise then fall)
5. Cinderella (rise then fall then rise)
6. Oedipus (fall then rise then fall)



Reagan et al. 2016 The emotional arcs of stories are dominated by six basic shapes

Sentiment analysis of storylines

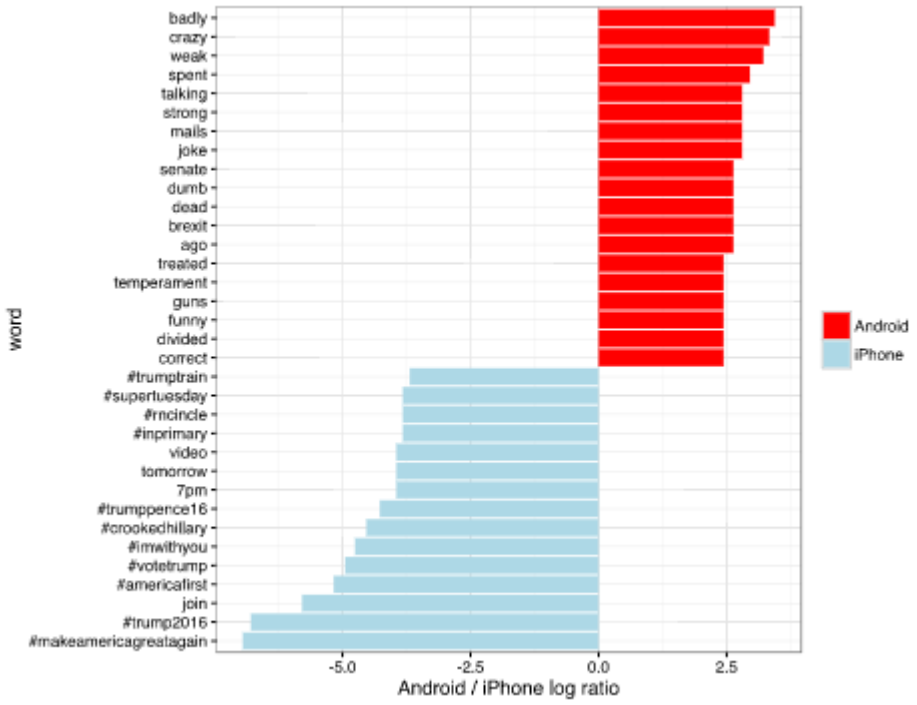
Words and phrases with known sentiment can be picked out in texts

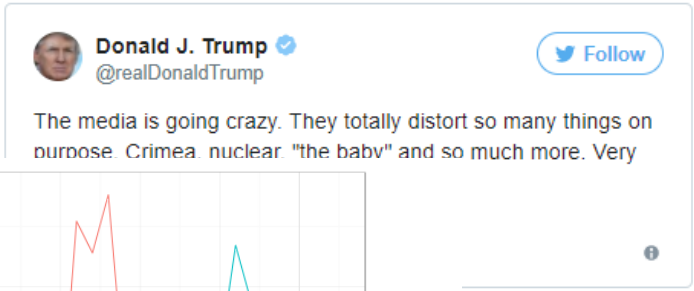


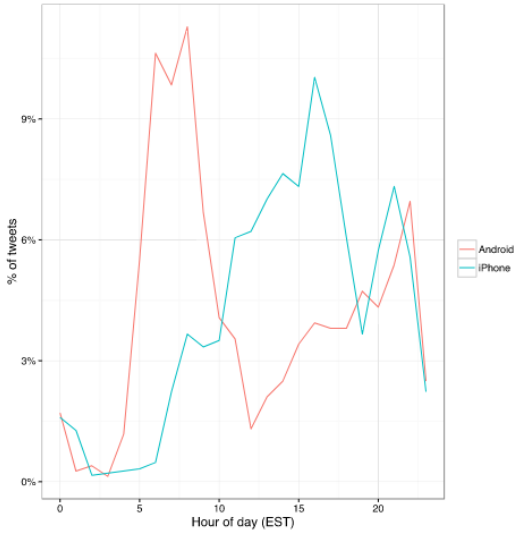
David Robinson 2017, Examining the arc of 100,000 stories: a tidy analysis

Text analysis of Trump's tweets confirms he writes only the (angrier) Android half

I don't normally post about politics (I'm not particularly savvy about polling, which is where data science has had the largest impact on politics). But this weekend I saw a hypothesis about Donald Trump's twitter account that simply begged to be investigated with data:







David Robinson, 2016, Trump Tweets

Birth of the cool

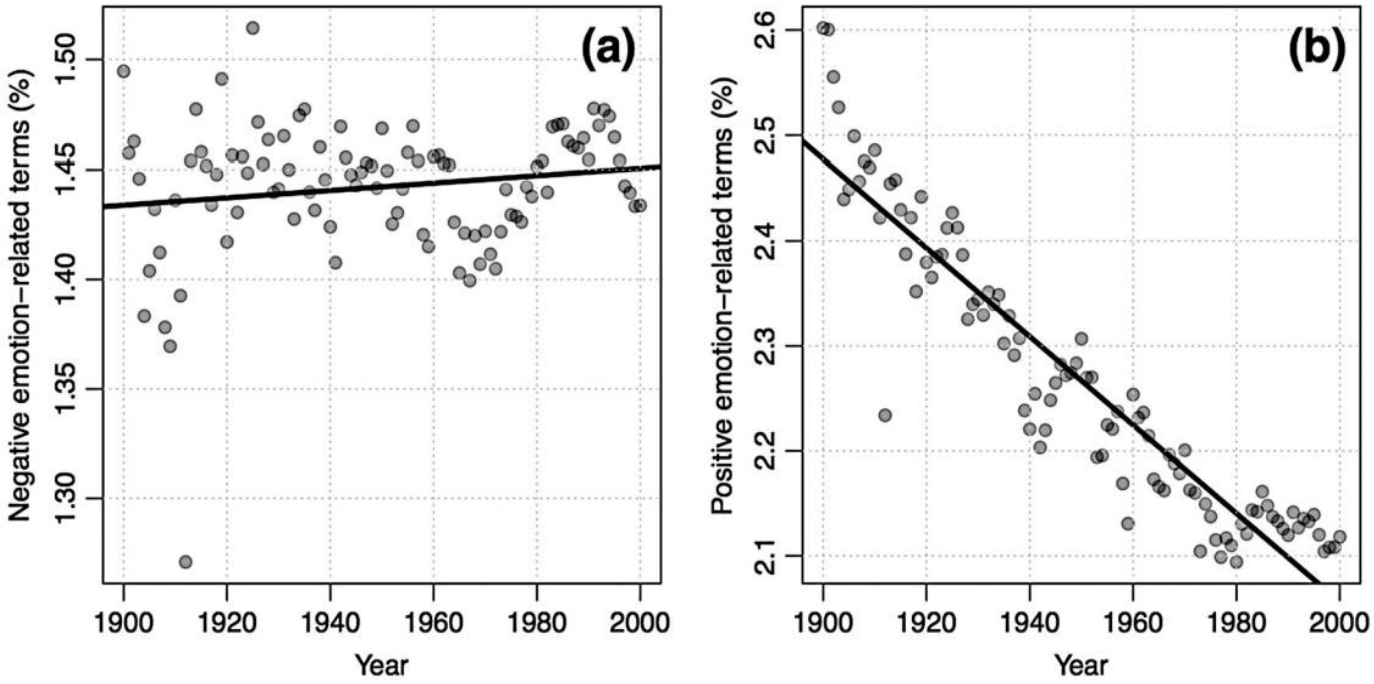


Figure 3. Emotionality changes in Anglophone literature, for the Google Books corpus. (a): negative emotions-related terms. (b): positive emotions-related terms. Solid lines represent linear regressions of the data.

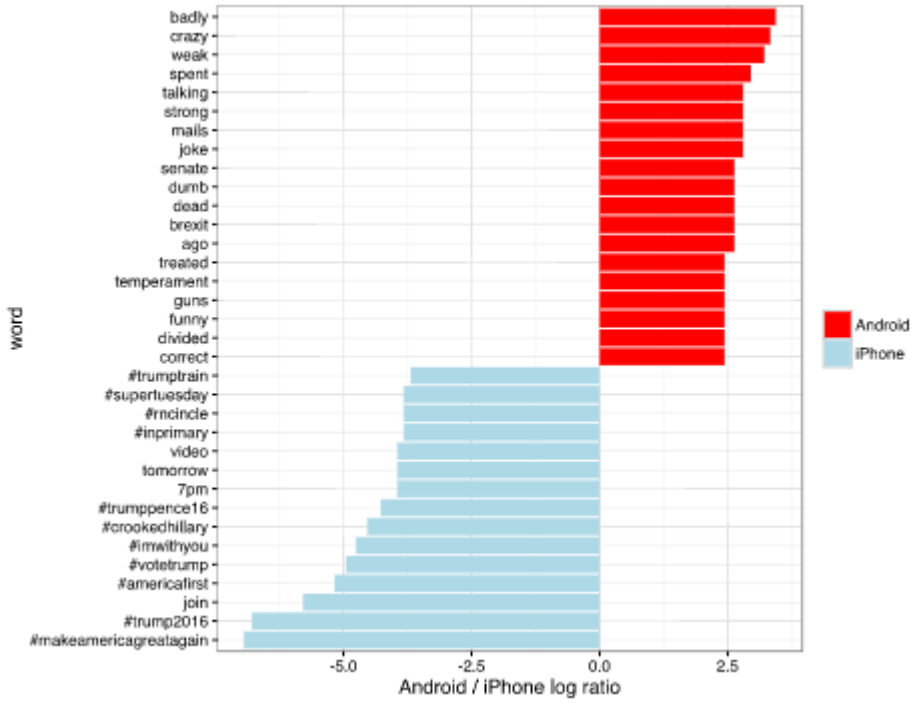
Morin & Acerbi 2016 Birth of the cool. *Cognition and Emotions*

Sentiment analysis in Lord of the Rings.

[Jakub Glinka 2017. Sentiment Analysis of the Lord of the Rings in tidytext]

Keywords in 2 groups of texts

E.g. Trump twitter Android & IPhone



Basic operations on texts

- **mutate(new_var = op(old_var) – make/change column**
- **filter(var==what_you_want) – get only the rows with right value**
- **count(var) – count unique values**
- **arrange(column) – sort by column**
- **arrange(desc(column) – reverse**
- **group_by(what)**
- **ungroup()**
- **rename() – just when needed**
- **summarize() – based on group_by**
- **unnest_tokens(words,line) – make lines into words**
- **str_detect(where, „what“) – check if it contains „what“**
- **anti_join(with_what, „var“) – remove matching values**
- **inner_join(with_what, „var“) – keep only matching values**
- **left_join(with_what, „var“) – just add where possible**
- **bind_df_idf(of, by, value) – get keywords**