



FLÜCHTLINGSKRISE SENTIMENT ANALYSIS

EMILY MARTIN

WHY THIS PROJECT?

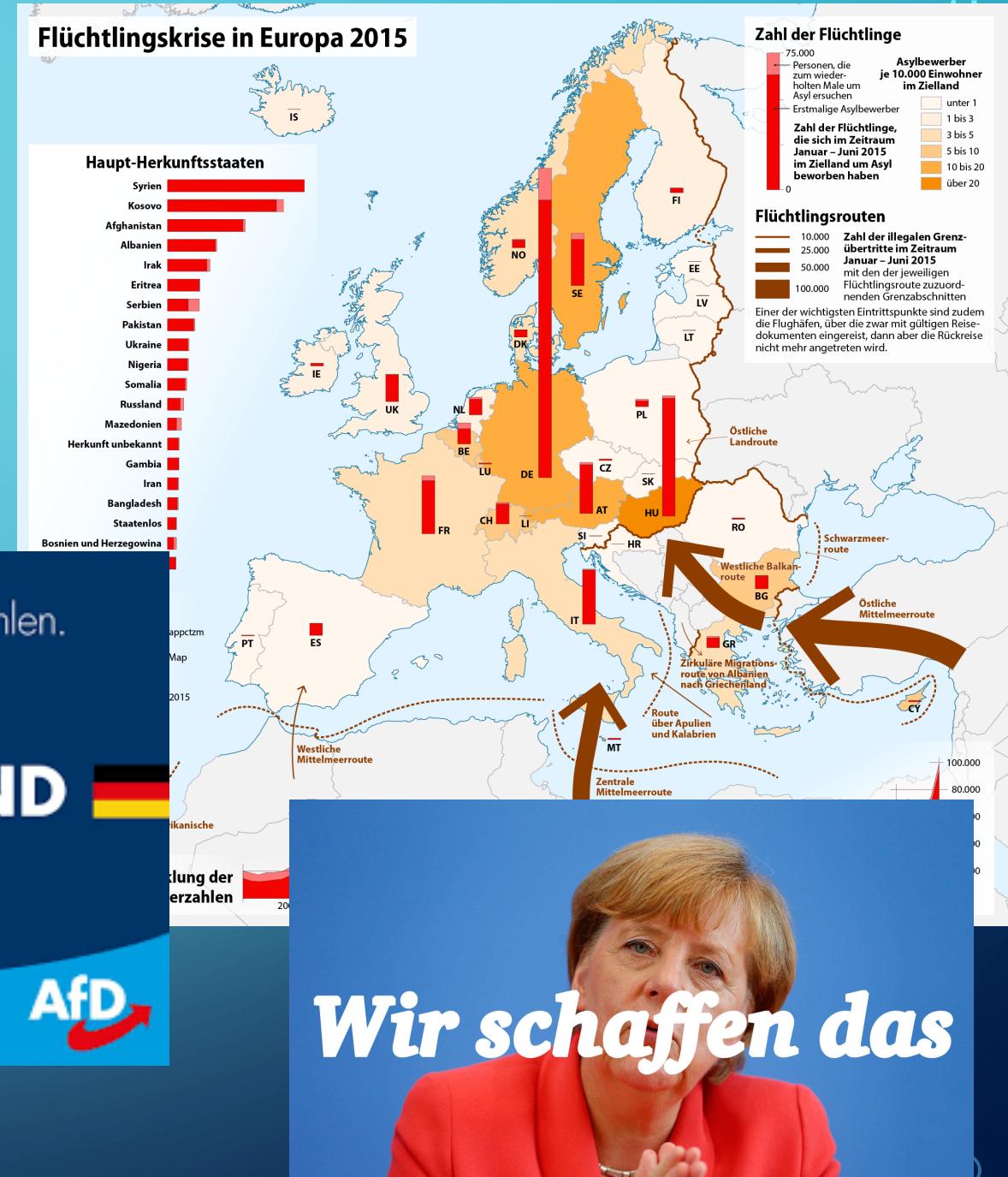


Am 24.9. **AfD** wählen.

HOL DIR DEIN LAND ZURÜCK!

Afd

PATRIOTISCHE EUROPÄER GEGEN ISLAMISIERUNG DES ABENDLANDES



WHAT WAS MY GOAL?

- Scrape newspapers across the political spectrum and do sentiments analysis to see if they differ in sentiments towards refugees.
- Possibilities beyond sentiment analysis:
 - Do different words for “refugee” carry different weights?
 - Does tense play any role?
 - German has so many “fun” tenses...

My Hypothesis

- More left leaning newspapers will report more positively about refugees than more right leaning newspapers.

- My sources:

- Die TAZ (Die Tageszeitung): left-wing/green; daily German newspaper with a modest circulation.
- Der Süddeutsche Zeitung: left-liberal, daily newspaper with a very wide circulation.
- Der Zeit: centrist/liberal; one of the largest weekly newspapers in Germany
- Junge Freiheit: strong right-wing leanings, small weekly newspaper.

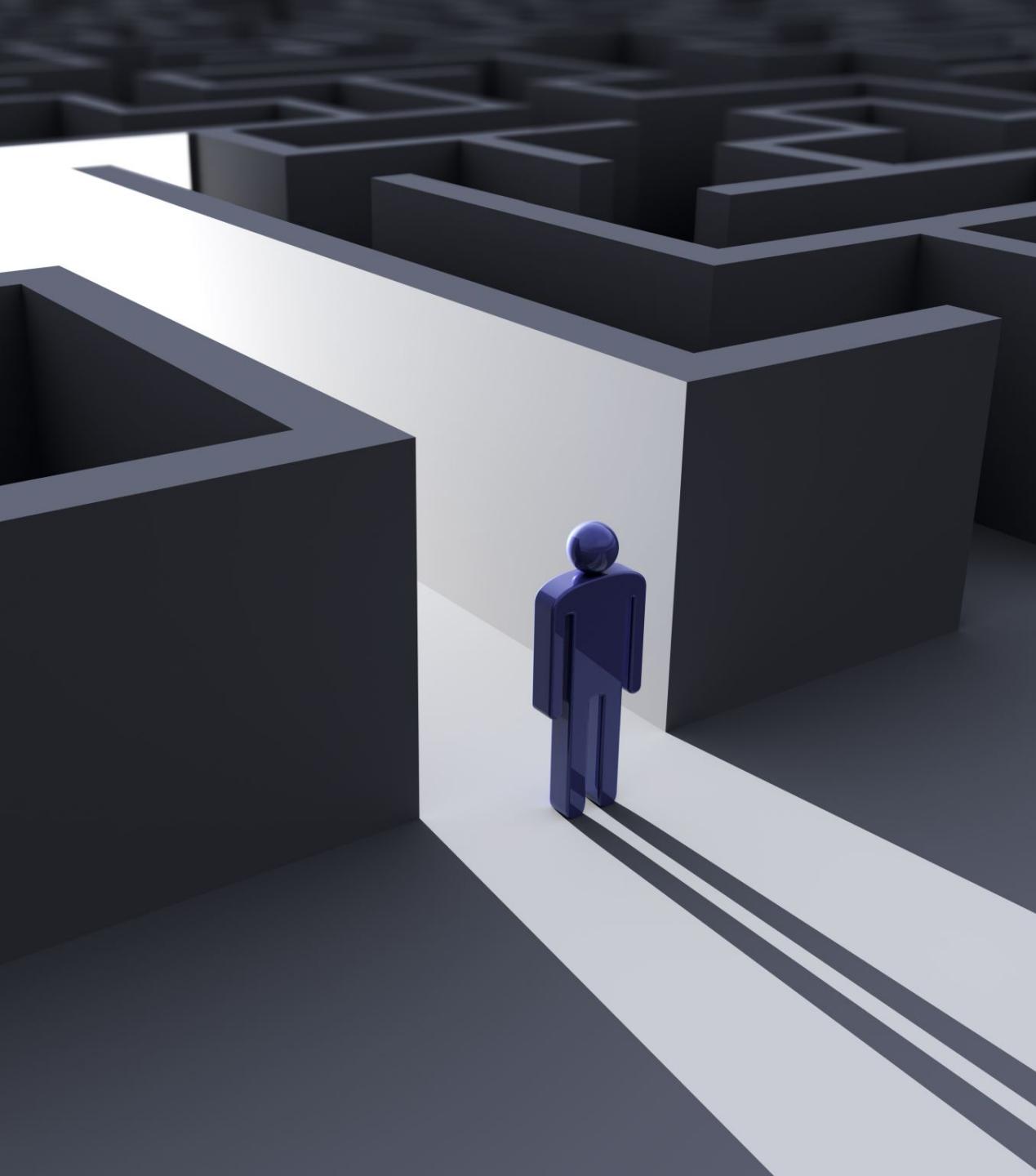
DATA ACQUISITION....

Web scraping is
easy!!

This won't take
long at all!

What are Na-Rae
and Joey so
worried about
anyway?





TURNS OUT IT IS NOT SO EASY...

- COPYRIGHT
- Every website is different!
 - So so different...
- “An API? Why have that?”
- Many different tools
 - How much data do you want?
 - How messy can it be?

THE SOLUTION!



- **Der Zeit**: Has an API! I collected links and dates, etc, through that, then used Urllib and Beautiful Soup to get the actual article text.
- **Die TAZ**: Pitt has access to their archive (can't scrape that), so I manually collected 100 links and then used Urllib and Beautiful Soup.
- **Der SZ**: I created 20 URLs of search result pages, then found each relevant link on the page and opened and scraped each one for date and text.
- **Junge Freiheit**: I opened every results page for the search term and filtered by date before collecting URLs to open and scrape.

```
> <div id="adzone_wall" class=""> ...</div>
-> <div id="pages" class="news no-support-columnspan"> event
  > <ul id="globalnavigation" class="navbar" role="navigation" style="overflow: visible;"> ...</ul>
  -> <div class="full news report article page first odd first_page n1" itemscope="" itemtype="http://schema.org/NewsArticle"> event
    -> <span class="body" role="main" style="min-height: 933px;">
      > <div id="xid819017" class="main rack first_rack">
        > <div class="metadata" xmlns="" itemscope=""> ...</div>
        -> <div id="" class="odd sect sect_article news report" role="region">
          -> <article class="sectbody" itemprop="articleBody">
            > <h1 xmlns="" itemprop="headline"> ...</h1>
            > <p class="intro " xmlns="" itemprop="description"> ...</p>
            > <a class="full picture" xmlns="" href="/picture/593993/948/FluechtlingsinTurnhalle.jpg" target="fullImage" onclick="var href = this.href; vHWin=window.open( href , 'fullIm...width=948,height=474' ); vHWin.focus(); return false;" itemprop="image"> ...</a> event
            > <p class="caption" xmlns=""> ...</p>
            > <p class="article first odd" xmlns=""> ...</p>
```

```
[48]: # Create a dictionary for dates and article texts with the link as the key.
# Use urllib and Beautiful Soup to open and get the text and dates from each link
date_dict = {}
art_dict_taz = {}
x = ''
for link in links:
    page = urllib.request.urlopen(link)
    soup = BeautifulSoup(page)
    t = soup.find("div", {"class": "main rack first_rack"})
    paras = t.findAll('p')
    a = [p.text for p in paras]
    d = soup.find("li", {"class": "date"})
    text = x.join(a)
    date_dict[link] = d
    art_dict_taz[link] = text

#art_dict_taz
#date_dict
```

A QUICK LOOK AT THE DATA

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1731 entries, 0 to 572
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   href         1731 non-null    object  
 1   text          1731 non-null    object  
 2   date          1731 non-null    object  
 3   word_count    1731 non-null    int64  
 4   sent_count    1731 non-null    int64  
 5   toks          1731 non-null    int64  
 6   types         1731 non-null    int64  
 7   TTR           1731 non-null    float64 
 8   source        1731 non-null    object  
dtypes: float64(1), int64(4), object(4)
memory usage: 135.2+ KB
```

```
Out[22]: SZ      982
Zeit    573
TAZ     100
JF      76
Name: source, dtype: int64
```

		href	text	date	word_count	sent_count	toks	types	TTR	source
245		http://www.zeit.de/kultur/2015-09/fluechtlings... ZEIT ONLINE: Frau Benhabib, die Flüchtlingskri...	ZEIT ONLINE: Frau Benhabib, die Flüchtlingskri...	2015-09-18T08:46:18Z	1092	77	1282	565	0.440718	Zeit
3		https://jungefreiheit.de/politik/deutschland/2... ERFURT. Asylbewerber, die mit der Deutschen Ba...	ERFURT. Asylbewerber, die mit der Deutschen Ba...	04. November 2015	191	12	227	157	0.691630	JF
38		https://jungefreiheit.de/sonderthema/2014/jf-t... Fast alle Medien sind sich einig: Die Dresdene...	Fast alle Medien sind sich einig: Die Dresdene...	12. Dezember 2014	446	46	537	275	0.512104	JF
290		https://www.sueddeutsche.de/muenchen/wolfratsh... Bald leben 1500 Flüchtlinge im Landkreis. Wie ...	Bald leben 1500 Flüchtlinge im Landkreis. Wie ...	27. November 2015, 19:01 Uhr	450	42	569	283	0.497364	SZ
484		https://www.sueddeutsche.de/panorama/friedrich... Die Narren im Südwesten sorgen sich wegen der ...	Die Narren im Südwesten sorgen sich wegen der ...	11. November 2015, 18:52 Uhr	121	10	149	101	0.677852	SZ
676		https://www.sueddeutsche.de/muenchen/muenchen-... Der Landrat wählt, womöglich bewusst, andere W...	Der Landrat wählt, womöglich bewusst, andere W...	26. Oktober 2015, 18:54 Uhr	477	23	567	318	0.560847	SZ
903		https://www.sueddeutsche.de/politik/migration-... Erfurt (dpa) - Begleitet von einem Großaufgebo...	Erfurt (dpa) - Begleitet von einem Großaufgebo...	8. Oktober 2015, 6:31 Uhr	93	6	103	80	0.776699	SZ
450		https://www.sueddeutsche.de/bayern/moosburg-cs... Nach dem fremdenfeindlichen Eklat im Ortsverba...	Nach dem fremdenfeindlichen Eklat im Ortsverba...	11. November 2015, 18:56 Uhr	287	20	334	211	0.631737	SZ

AT LAST... ANALYSIS!

- Why SpaCy?
 - Language processing pipelines, easy to build and customize.
 - SpaCy speaks German!
 - Someone built a package for sentiment analysis on German
 - Sentiws (<https://spacy.io/universe/project/spacy-sentiws>)

```
# An example of how sentiws works and a look at the pipeline
for token in doc:
    print('{}, {}, {}'.format(token.text, token._.sentiws, token.pos_))

print(nlp.pipeline)

Die, None, DET
Dummheit, -0.4877, NOUN
der, None, DET
Unterwerfung, -0.3279, NOUN
blüht, 0.2028, VERB
in, None, ADP
häbschen, 0.4629, ADJ
Farben, None, NOUN
., None, PUNCT
[('tok2vec', <spacy.pipeline.tok2vec.Tok2Vec object at 0x7fb1b32e5710>), ('tagger', <spacy.pipeline.tagger.Tagger object at 0x7fb1b32f5ef0>), ('morphologizer', <spacy.pipeline.morphologizer.Morphologizer object at 0x7fb1b3305d70>),
 , ('parser', <spacy.pipeline.dep_parser.DependencyParser object at 0x7fb1b316ac20>), ('attribute_ruler', <spacy.pipeline.attributeruler.AttributeRuler object at 0x7fb1b331ec30>), ('lemmatizer', <spacy.pipeline.lemmatizer.Lemmatizer object at 0x7fb1b33a6f00>), ('sentiws', <function my_component at 0x7fb1b31af680>)]
```

GETTING THE SENTIMENT SCORES

The process:

- Passed each text through the pipeline
- Removed stop words
- If the word had a sentiws sentiment score I collected it
- Took the mean of the scores for each article (why not the median?)
- If it was <0 -> ‘neg’, if it was >0 -> ‘pos’

Some very important notes:

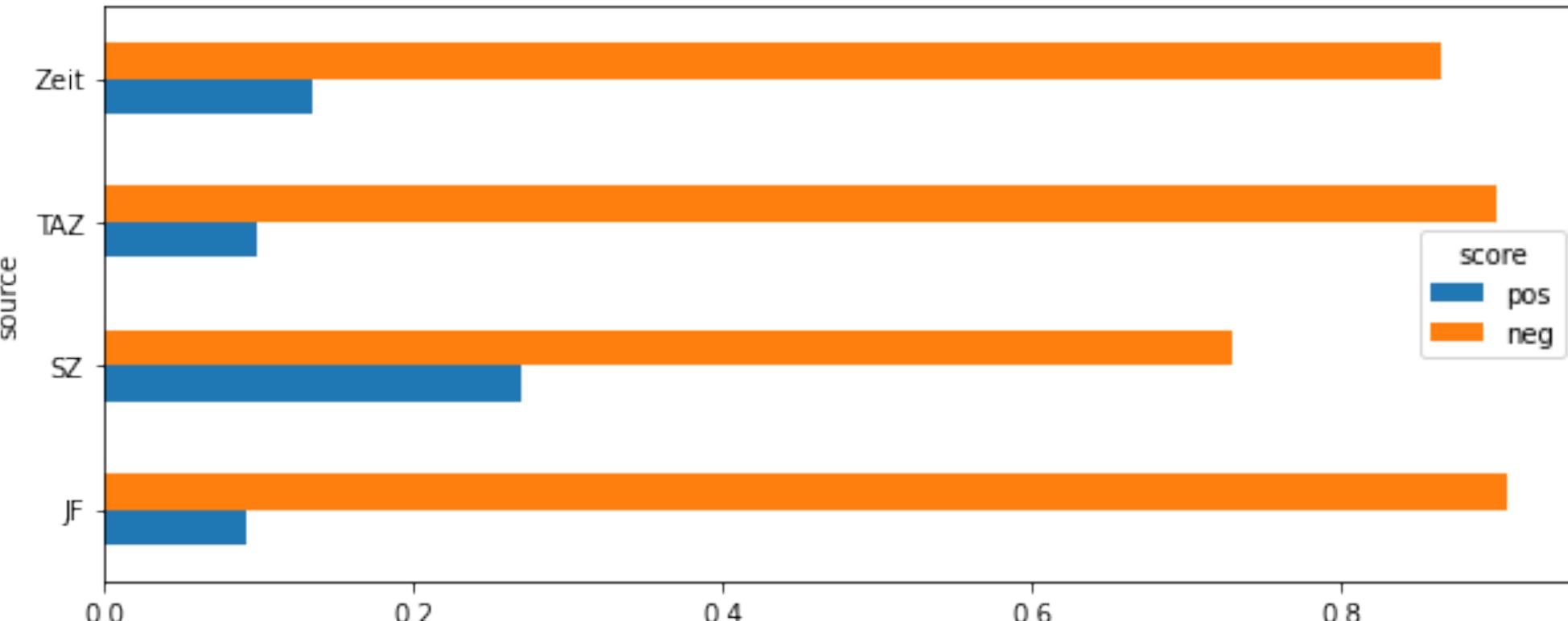
- Flüchtling (refugee) has a negative weight (-0.0048) and it is in all my articles
 - Interesting in its own right
 - Most words have a score of ‘None’
 - ‘Migrant’ (migrant/another term often used) does not have a weight from sentiws

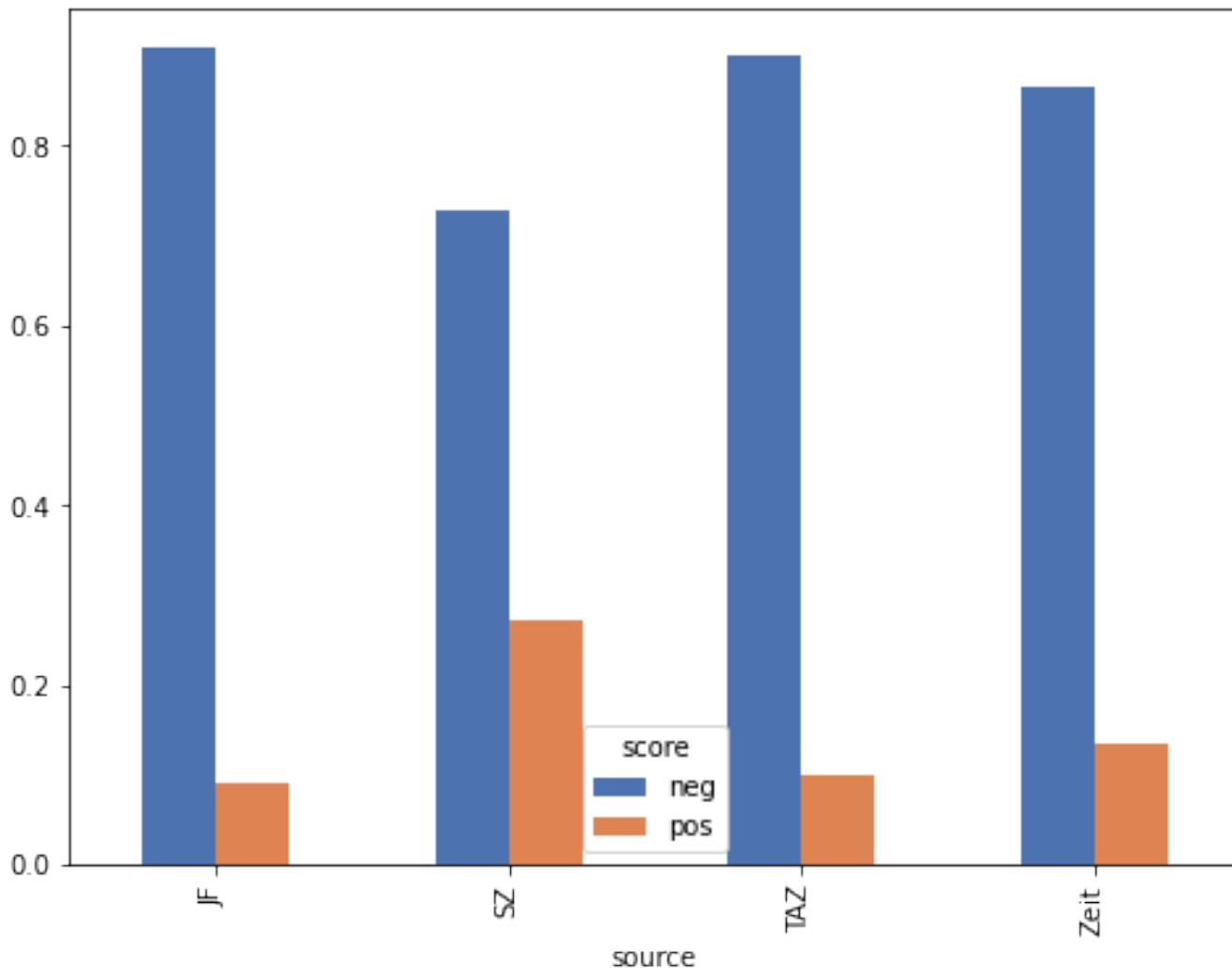


BUT WHAT DID I FIND??

YEAH, GET TO SOME PLOTS ALREADY...

Sentiment by Source

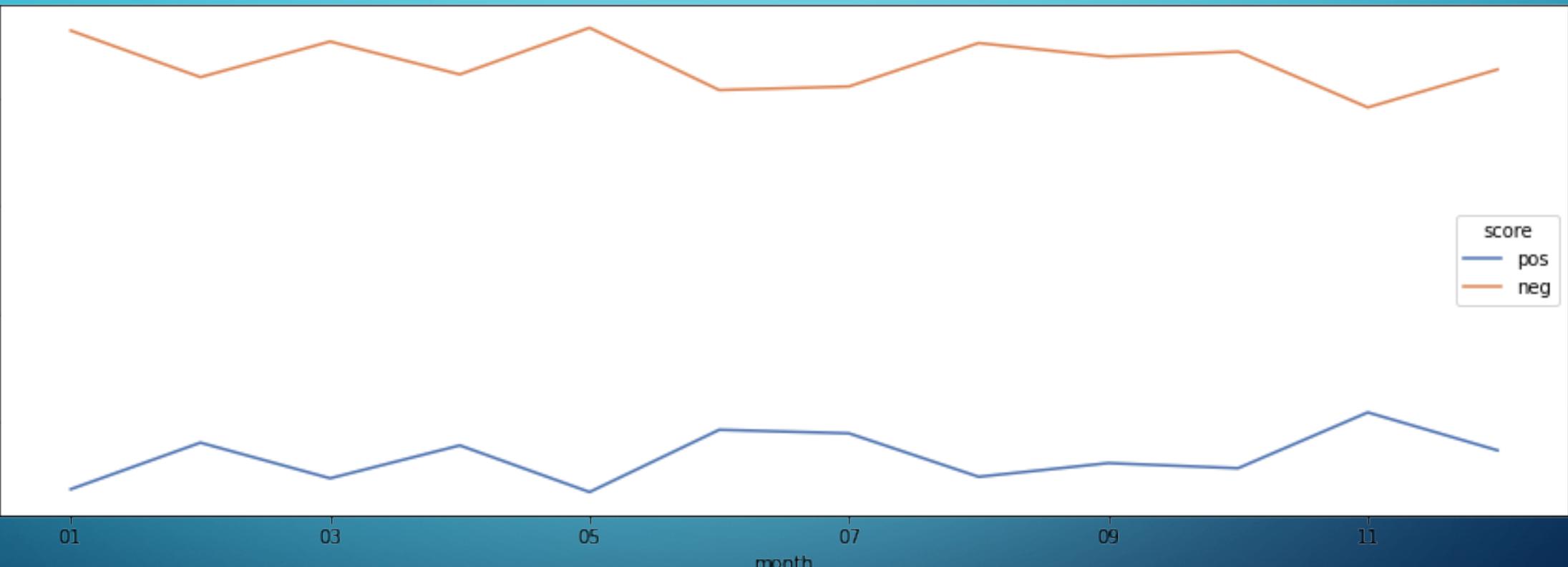




```
source    score
JF        neg      69
          pos      7
SZ        neg     711
          pos     264
TAZ       neg      90
          pos     10
Zeit      neg     493
          pos     77
Name: score, dtype: int64
```

```
source    score
JF        neg    0.907895
          pos    0.092105
SZ        neg    0.729231
          pos    0.270769
TAZ       neg    0.900000
          pos    0.100000
Zeit      neg    0.864912
          pos    0.135088
Name: score, dtype: float64
```

Change in sentiment by month for der Zeit



NOTES/FURTHER GOALS

I willingly acknowledge there are flaws and room for improvement in this project

- Sentiws may not have been the best tool
- The scraping scripts are temperamental
- Uneven distribution of data and dates of publication
- Data cleaning...

Future goals/possibilities

- Sentence based analysis
- Tenses?
- Over-time analysis for more sources
- Try another sentiment analysis tool?



ANY QUESTIONS?

THANK YOU ALL ☺

A BIG thank you to everyone who helped me: @Ruth Martin, @Na-Rae, @Joey