Data Analytics

CS301

Text Analysis:

Sentiment Determination

Fall 2018

Oliver Bonham-Carter

Exam 2

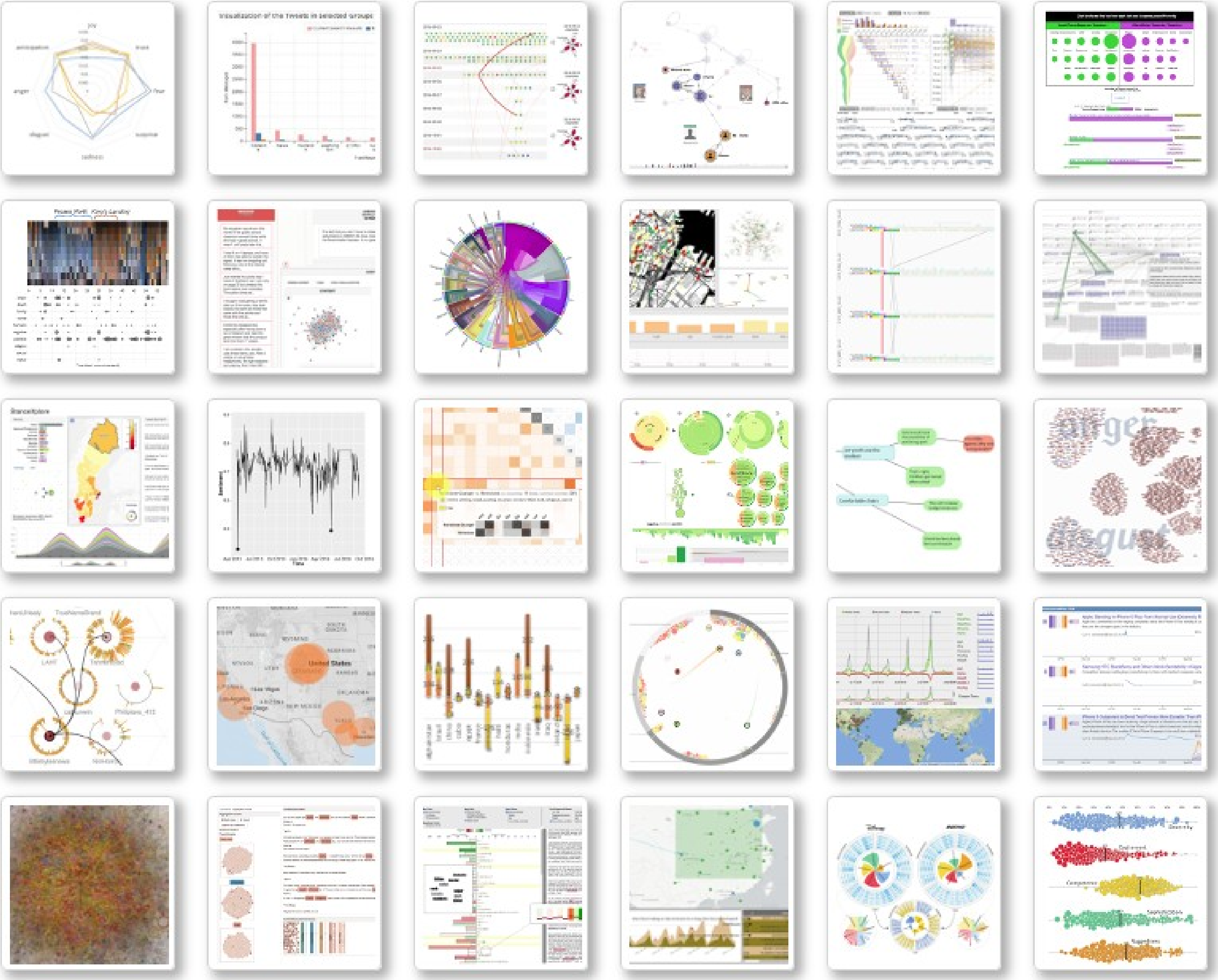
- During class: Friday, 30 Nov 2018
- Multiple choice, short answer, T/F, matching
- Study slides and then go to book for detail.
- Interpret code or predict its outcome.
- Topics include
  - Relational data frames: types of joins
  - Factors and uses
  - Function syntax: recognizing functions that work
  - models
    - t.test
    - Linear regression: uses, assumptions and interpretations
    - Hypotheses and Statistics from regression and t-tests tests
    - Correlations
    - Code in R to do these tests. Find bugs
  - Text mining: general uses and steps in analysis.

Text Analysis: Sentiment of Content

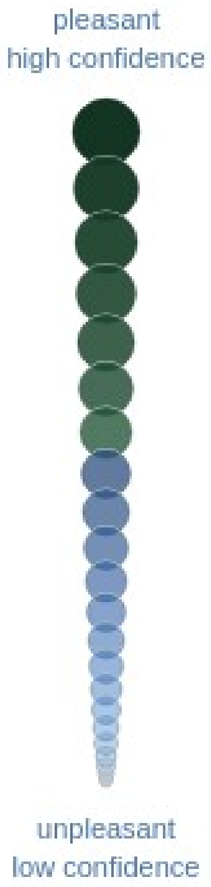
- The determination of the text’s “message” or “mood” based on the actual individual *words*.
- How good, how bad is the writer feeling about some topic?
- Is a body of text describing some idea where many of the words are emotionally charged with some type of feeling?
- Sentiment analysis is able to determine what the general feeling is behind some written work.

Visualizing Schemes are being developed

- To find out about new work in visualizing analytics, check out the SentimentVis Browser at http://sentimentvis.lnu.se/



Online tool: Sentiment Viz



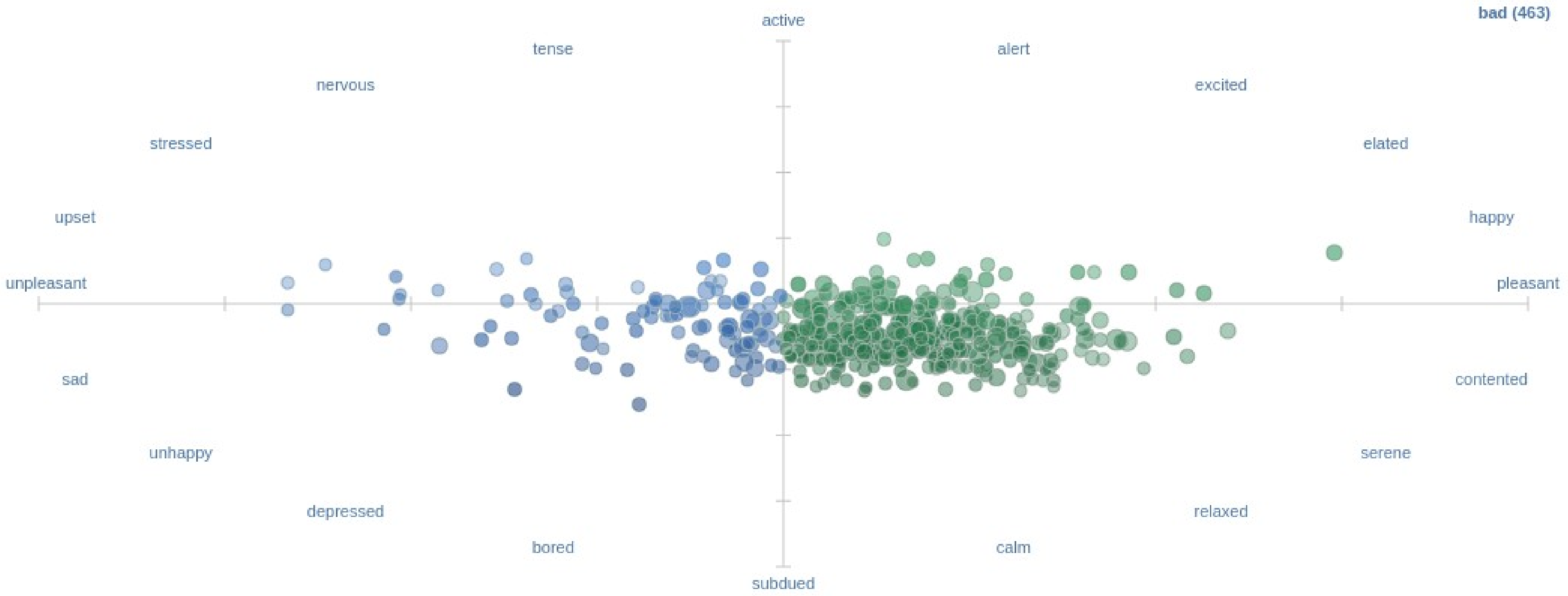
https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

What Is?

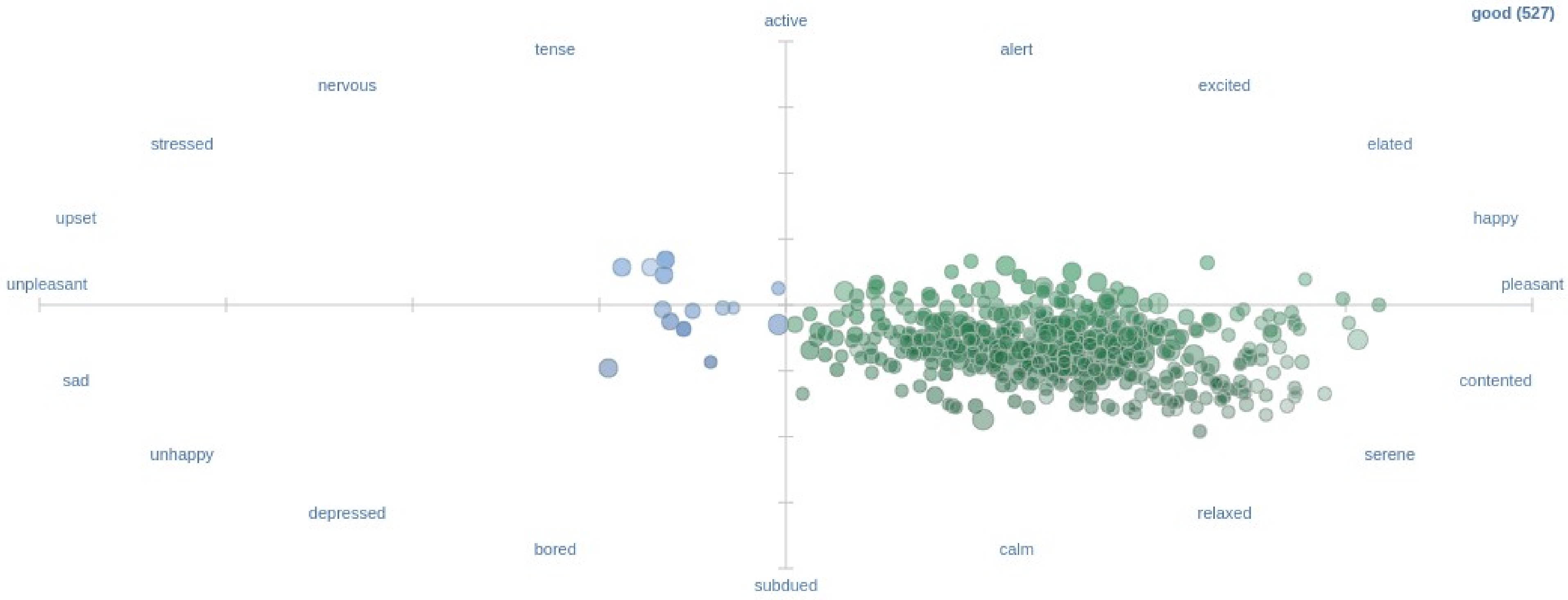
- User-entered keywords are parsed in the tweets of the day.
- Tweets are presented using several different visualization techniques. Each technique is designed to highlight different aspects of the tweets and their sentiment.
- The sentiment tab visualizes where tweets lie in an emotional scatterplot with pleasure and arousal on its horizontal and vertical axes.
- The spatial distribution of the tweets summarizes their overall sentiment.
- The number of queries per minute is limited...

https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

The word, “Bad”

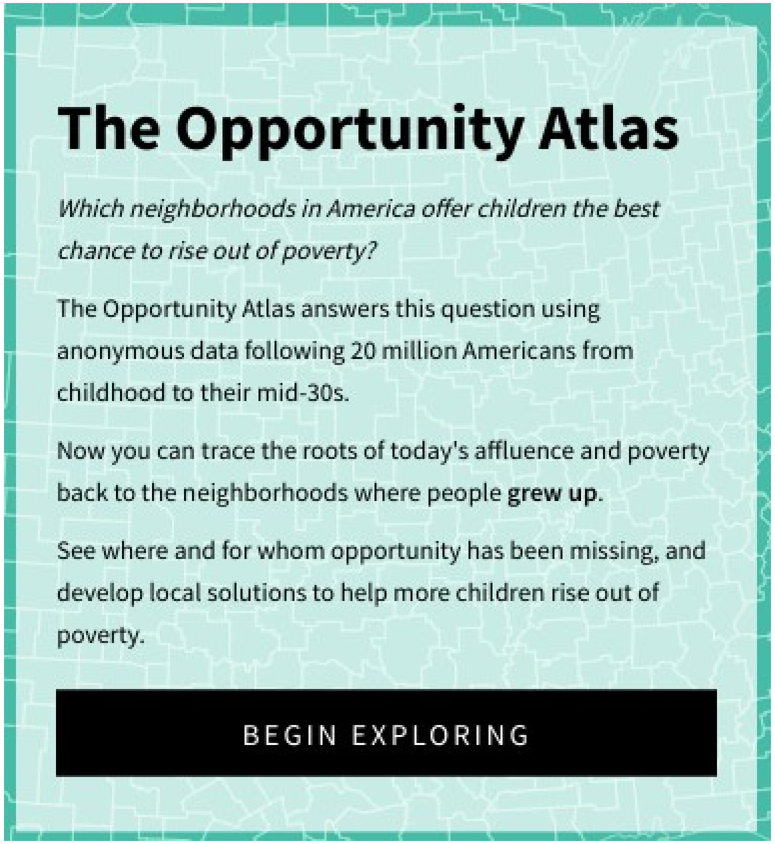

bad (463)
active
tense
alert
nervous
excited
stressed
elated
upset
happy
unpleasant
pleasant
sad
contented
unhappy
serene
depressed
relaxed
bored
calm
subdued

The word, “Good”



Click around on the web site to discover new ways of viewing data.

Online Tool:
The Opportunity Atlas



The Opportunity Atlas

Which neighborhoods in America offer children the best chance to rise out of poverty?

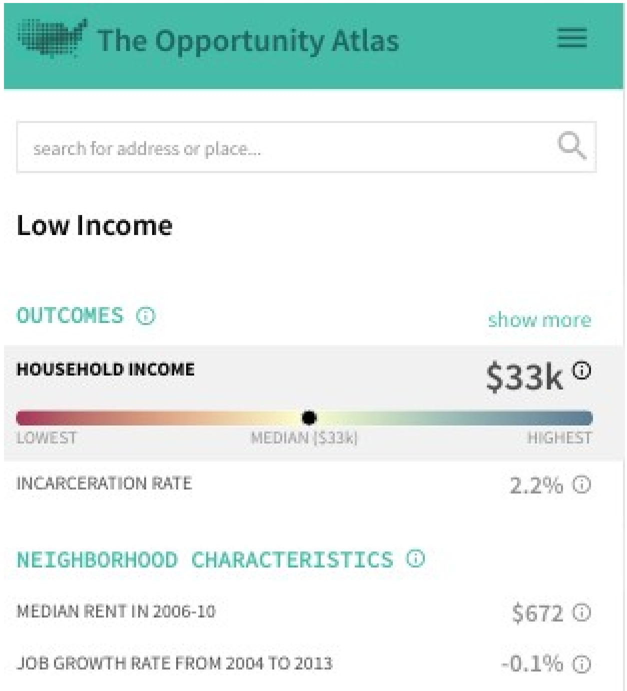
The Opportunity Atlas answers this question using anonymous data following 20 million Americans from childhood to their mid-30s.

Now you can trace the roots of today's affluence and poverty back to the neighborhoods where people **grew up**.

See where and for whom opportunity has been missing, and develop local solutions to help more children rise out of poverty.

https://www.opportunityatlas.org/

Online Tool:
The Opportunity Atlas



The Opportunity Atlas

search for address or place...

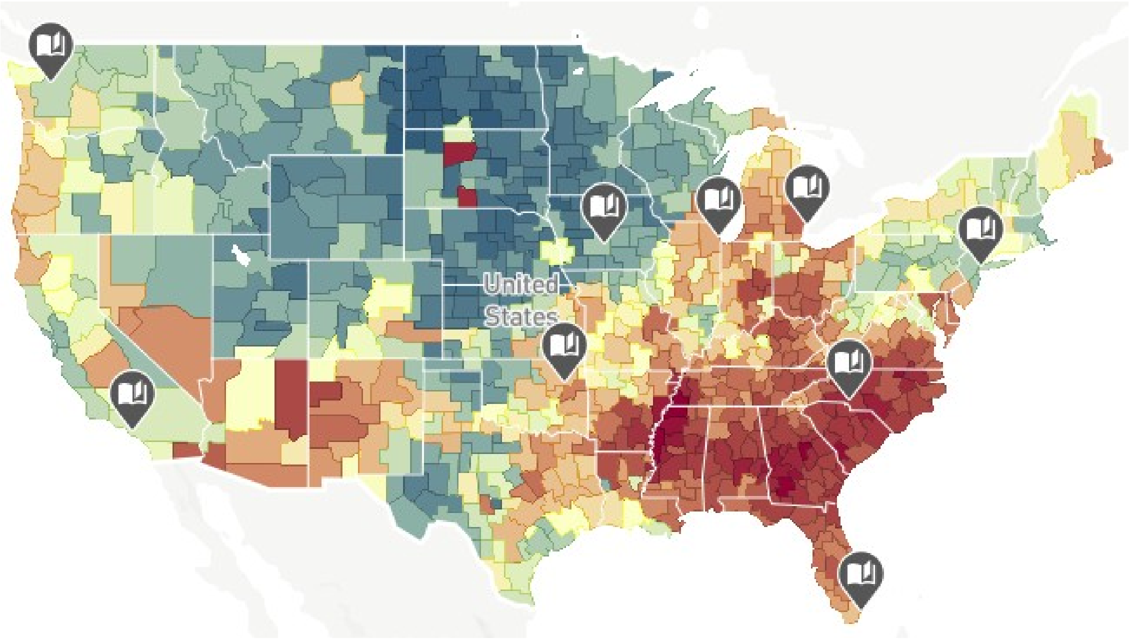
Low Income

OUTCOMES ⓘ show more

HOUSEHOLD INCOME	$33k ⓘ
LOWEST MEDIAN ($33k) HIGHEST	
INCARCERATION RATE	2.2% ⓘ

NEIGHBORHOOD CHARACTERISTICS ⓘ

MEDIAN RENT IN 2006-10	$672 ⓘ
JOB GROWTH RATE FROM 2004 TO 2013	-0.1% ⓘ

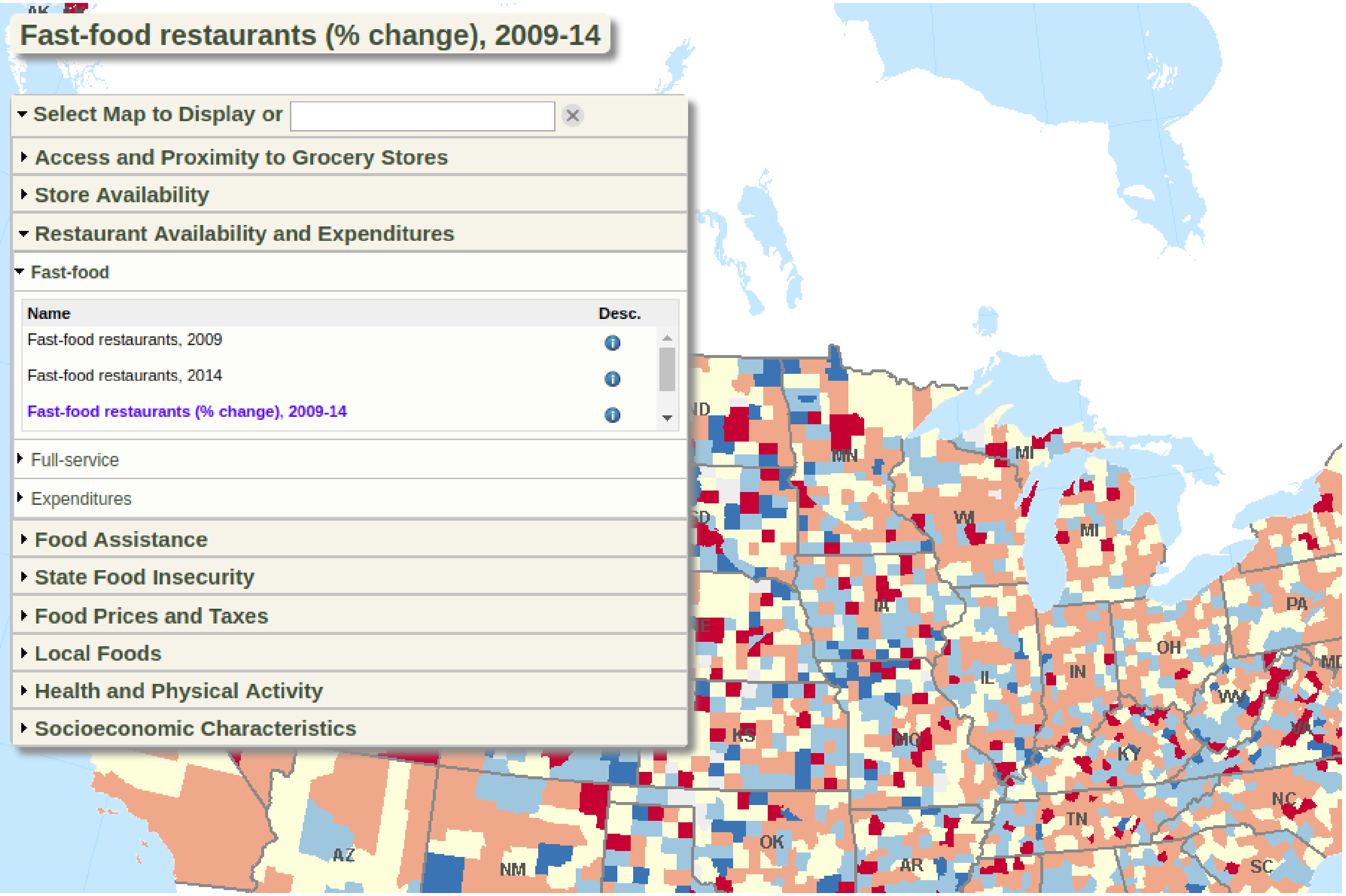


Determine the statistical amount of opportunity for careers, educational development and similar by a map.

https://www.opportunityatlas.org/

Online Tool:
The US Dept of Agriculture

Go to the Atlas



Fast-food restaurants (% change), 2009-14

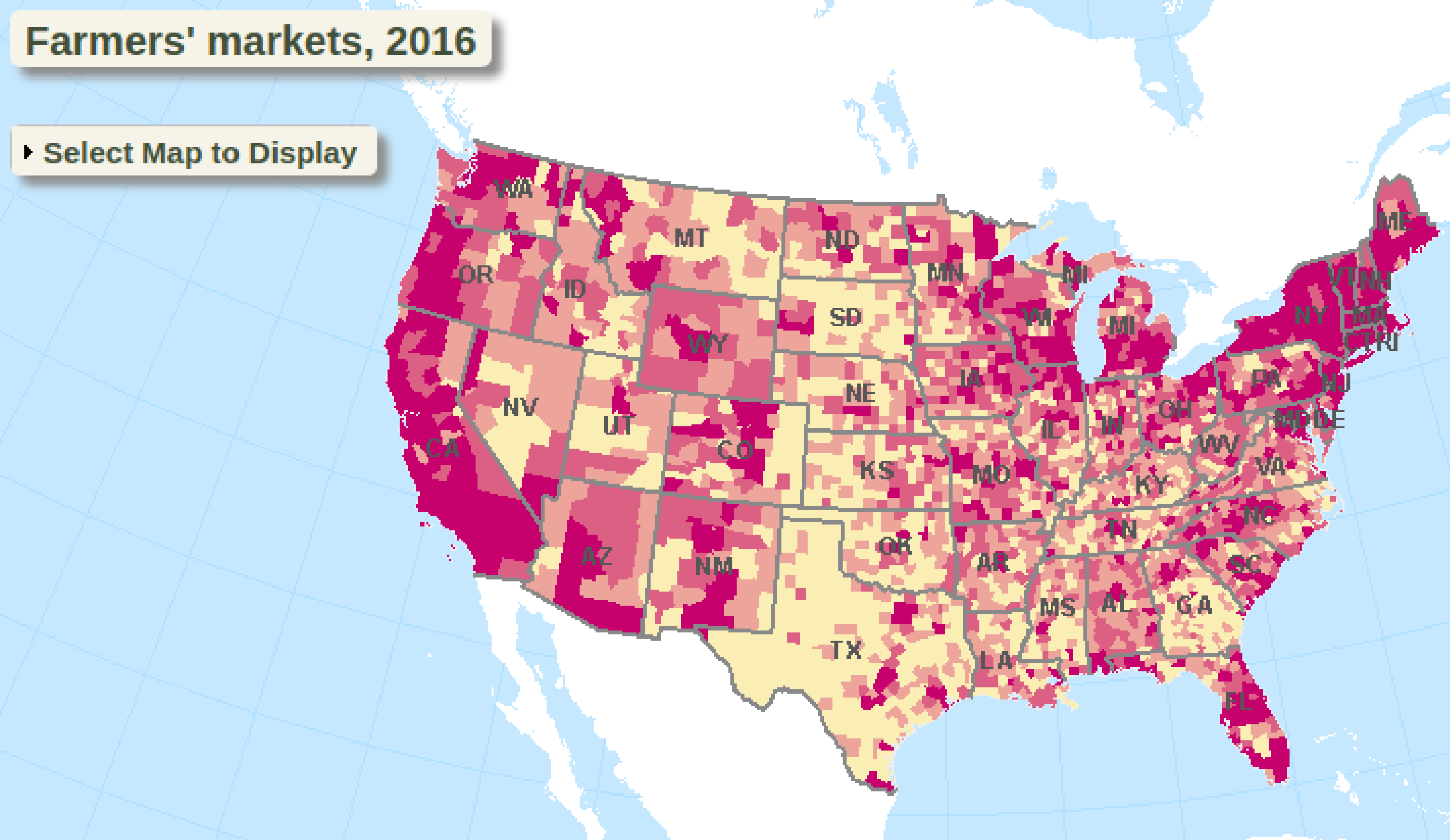
- Select Map to Display or
- Access and Proximity to Grocery Stores
- Store Availability
- Restaurant Availability and Expenditures
  - Fast-food

Name	Desc.
Fast-food restaurants, 2009	ⓘ
Fast-food restaurants, 2014	ⓘ
Fast-food restaurants (% change), 2009-14	ⓘ

  - Full-service
  - Expenditures
- Food Assistance
- State Food Insecurity
- Food Prices and Taxes
- Local Foods
- Health and Physical Activity
- Socioeconomic Characteristics

https://www.ers.usda.gov/data-products/food-environment-atlas/go-to-the-atlas/

Online Tool:
The US Dept of Agriculture



Mapping the number of Farmer’s Markets available in 2016

https://www.ers.usda.gov/data-products/food-environment-atlas/go-to-the-atlas/

Online Tool: The Institute for Health Metrics and Evaluation

IHME

Measuring what matters

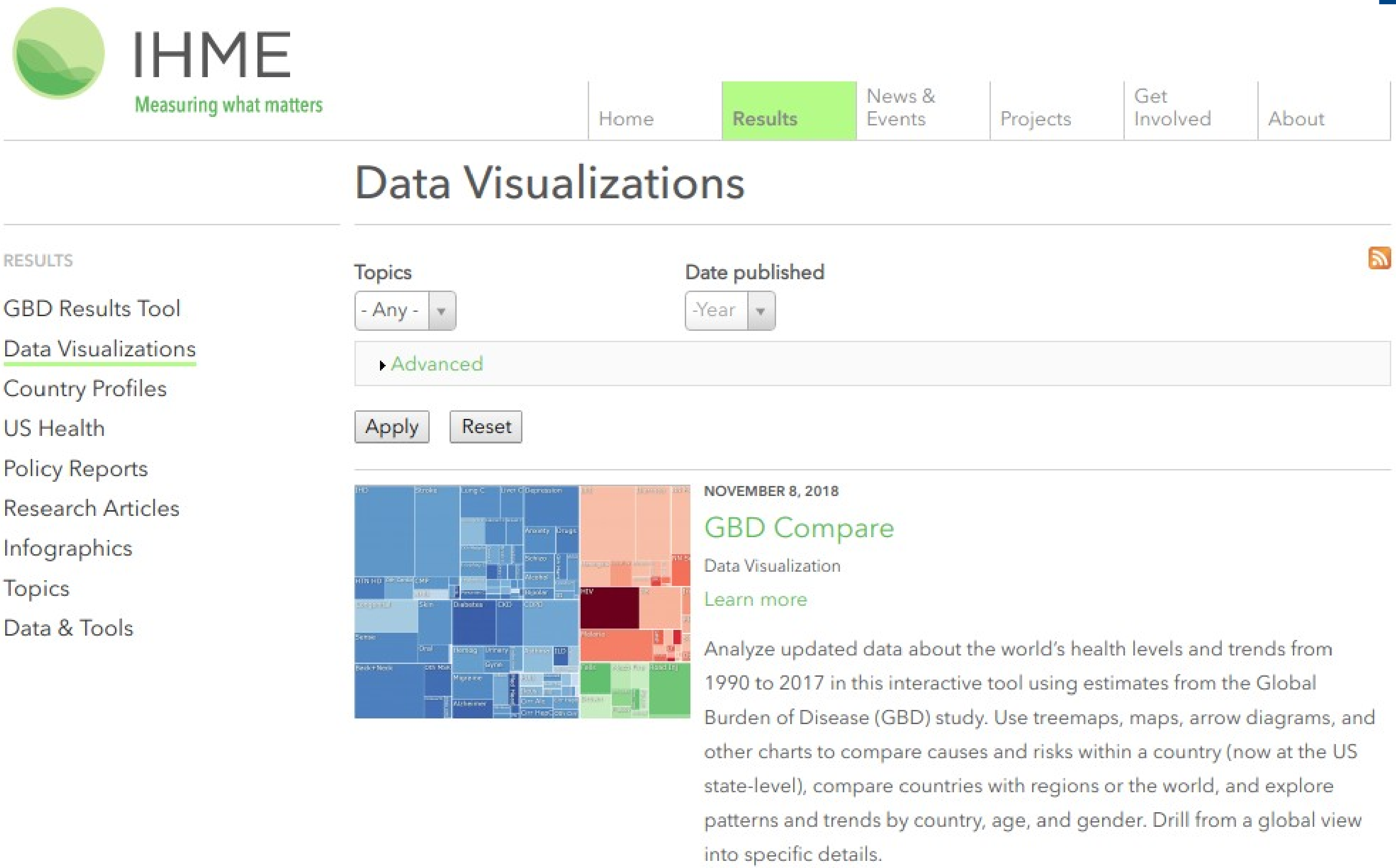
New neurology studies a ‘wakeup call’ for global health

Photo by Alvin Gogineni, Genentech.

http://www.healthdata.org/

https://vizhub.healthdata.org/epi/

Online Tool: The Institute for Health Metrics and Evaluation



Data Visualizations

NOVEMBER 8, 2018

GBD Compare

Data Visualization

Learn more

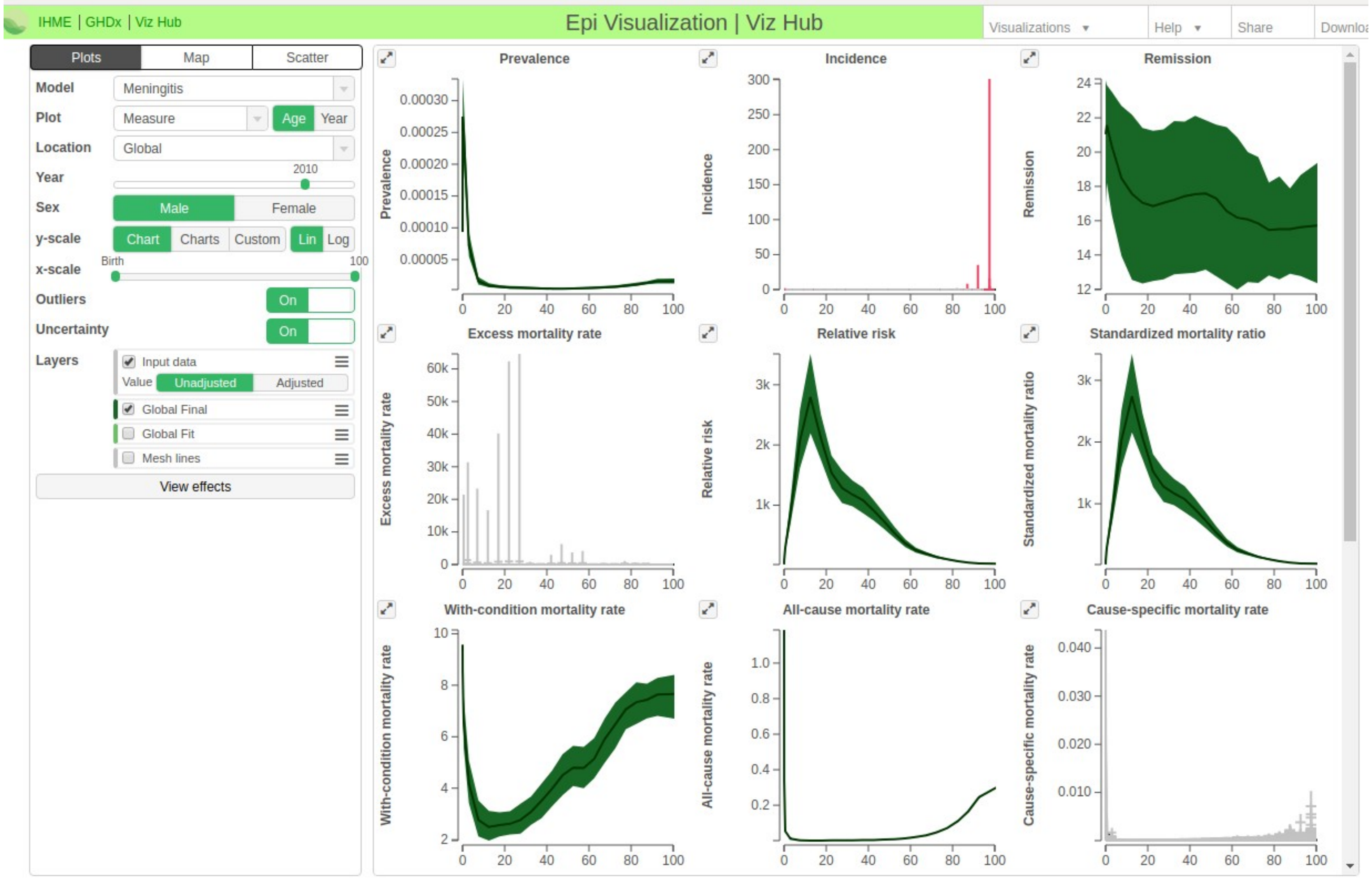
Analyze updated data about the world's health levels and trends from 1990 to 2017 in this interactive tool using estimates from the Global Burden of Disease (GBD) study. Use treemaps, maps, arrow diagrams, and other charts to compare causes and risks within a country (now at the US state-level), compare countries with regions or the world, and explore patterns and trends by country, age, and gender. Drill from a global view into specific details.

Visualize data on seemingly any topic of health

http://www.healthdata.org/

https://vizhub.healthdata.org/epi/

Online Tool: The Institute for Health Metrics and Evaluation



Model: Meningitis

Plot: Measure; Age | Year

Location: Global

Year: 2010

Sex: Male | Female

y-scale: Chart | Charts | Custom; Lin | Log

x-scale: Birth – 100

Outliers: On

Uncertainty: On

Layers:
- ☑ Input data — Value: Unadjusted | Adjusted
- ☑ Global Final
- □ Global Fit
- □ Mesh lines

https://vizhub.healthdata.org/epi/

Packages and Libraries

```
# install.packages(“janeaustenr")
# install.packages(“stringr”)

library(janeaustenr)
library(dplyr)
library(stringr)
library(tidyverse)
```

Data: Jane Austen’s Text

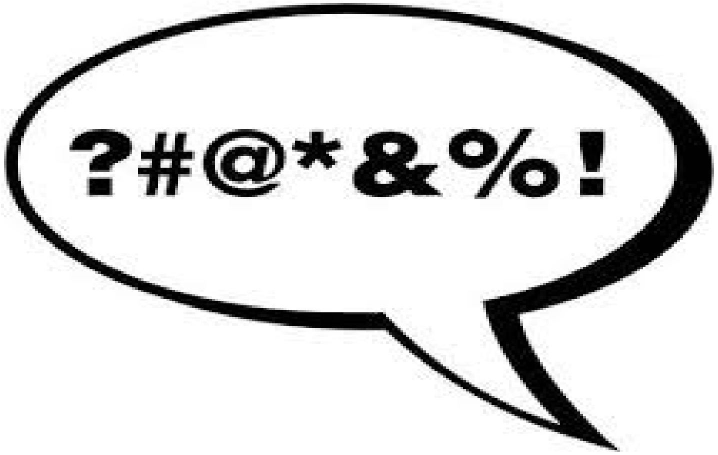
- Jane Austen’s 6 completed, published novels from the *janeaustenr* package.
  - Sense & Sensibility
  - Pride & Prejudice
  - Mansfield Park
  - Emma
  - Northanger Abbey
  - Persuasion

Research Question

- Jane Austen’s written work:

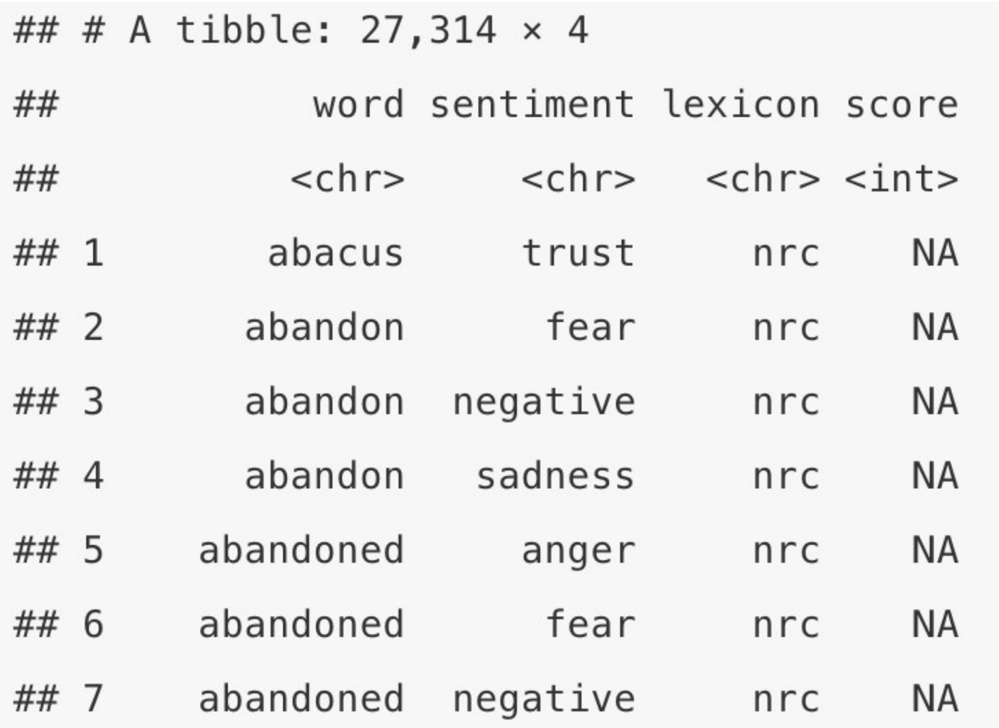
  How many *Bad* words did she use?

  How many *Good* words did she use?



The *Sentiments* dataset

```
install.packages("tidytext")
library(tidytext)
sentiments
```



```
## # A tibble: 27,314 × 4
##             word sentiment lexicon score
##            <chr>     <chr>   <chr> <int>
## 1         abacus     trust     nrc    NA
## 2        abandon      fear     nrc    NA
## 3        abandon  negative     nrc    NA
## 4        abandon   sadness     nrc    NA
## 5      abandoned     anger     nrc    NA
## 6      abandoned      fear     nrc    NA
## 7      abandoned  negative     nrc    NA
```

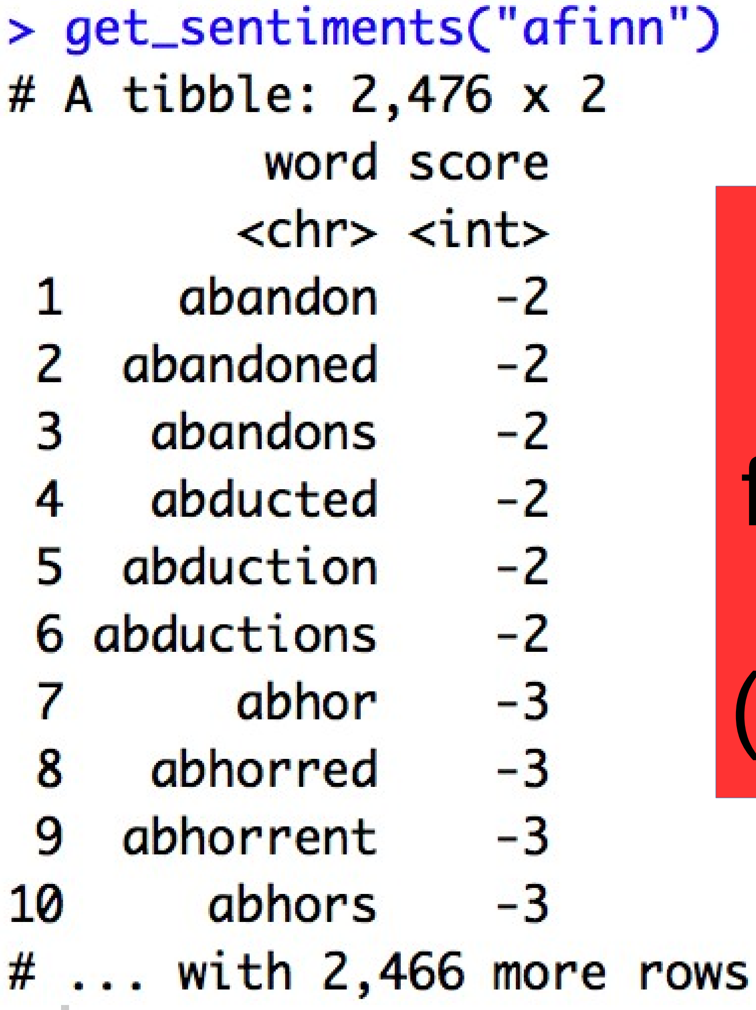
Three general-purpose lexicons

- ***AFINN*** from Finn Årup Nielsen,
  - assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment
- ***bing*** from Bing Liu and collaborators,
  - categorizes words in a binary fashion into positive and negative categories
- ***nrc*** from Saif Mohammad and Peter Turney
  - categorizes words in a binary fashion (“yes”/“no”) into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

- Used to determine the general mood of words.
- Lexicons are based on unigrams, (i.e., single words).
- Words are assigned scores for positive/negative sentiment,
- Emotions: joy, anger, sadness and etc.

Sentiments: **afinn**

- get_sentiments("afinn")

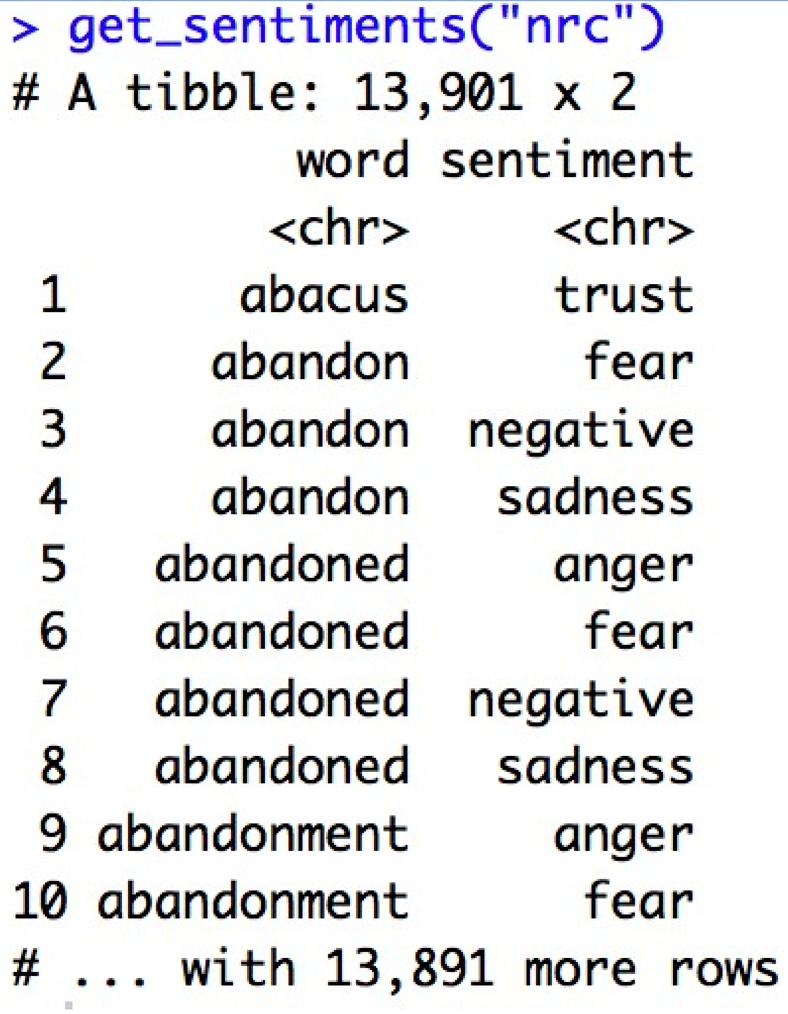


```
> get_sentiments("afinn")
# A tibble: 2,476 x 2
          word score
         <chr> <int>
 1     abandon    -2
 2   abandoned    -2
 3    abandons    -2
 4    abducted    -2
 5   abduction    -2
 6 abductions    -2
 7       abhor    -3
 8    abhorred    -3
 9   abhorrent    -3
10      abhors    -3
# ... with 2,466 more rows
```

Returns
a score
for each word
[-5, 5]
(Bad to Good)

Sentiments: **nrc**

- get_sentiments("nrc")

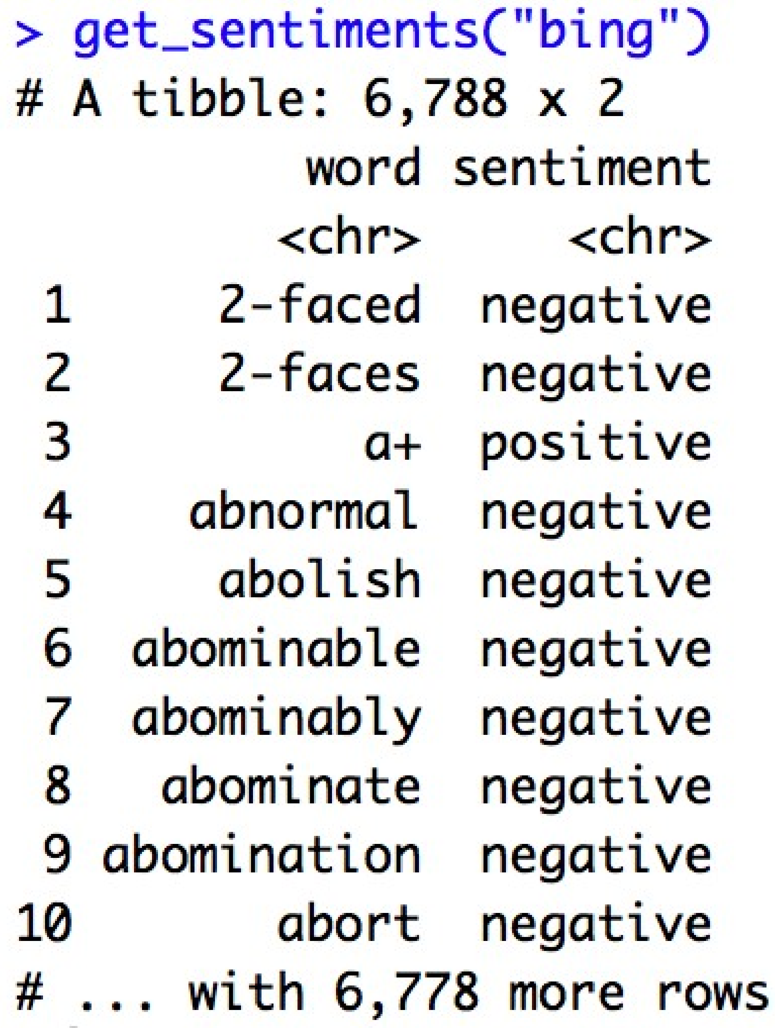


```
> get_sentiments("nrc")
# A tibble: 13,901 x 2
          word sentiment
         <chr>     <chr>
 1      abacus     trust
 2     abandon      fear
 3     abandon  negative
 4     abandon   sadness
 5   abandoned     anger
 6   abandoned      fear
 7   abandoned  negative
 8   abandoned   sadness
 9 abandonment     anger
10 abandonment      fear
# ... with 13,891 more rows
```

Returns
a *synonym*
for each word

Sentiments: **bing**

get_sentiments("bing")



```
> get_sentiments("bing")
# A tibble: 6,788 x 2
          word sentiment
         <chr>     <chr>
 1     2-faced  negative
 2     2-faces  negative
 3          a+  positive
 4    abnormal  negative
 5     abolish  negative
 6  abominable  negative
 7  abominably  negative
 8   abominate  negative
 9 abomination  negative
10       abort  negative
# ... with 6,778 more rows
```

Returns
a Positive
or
a Negative
measurement
for each word

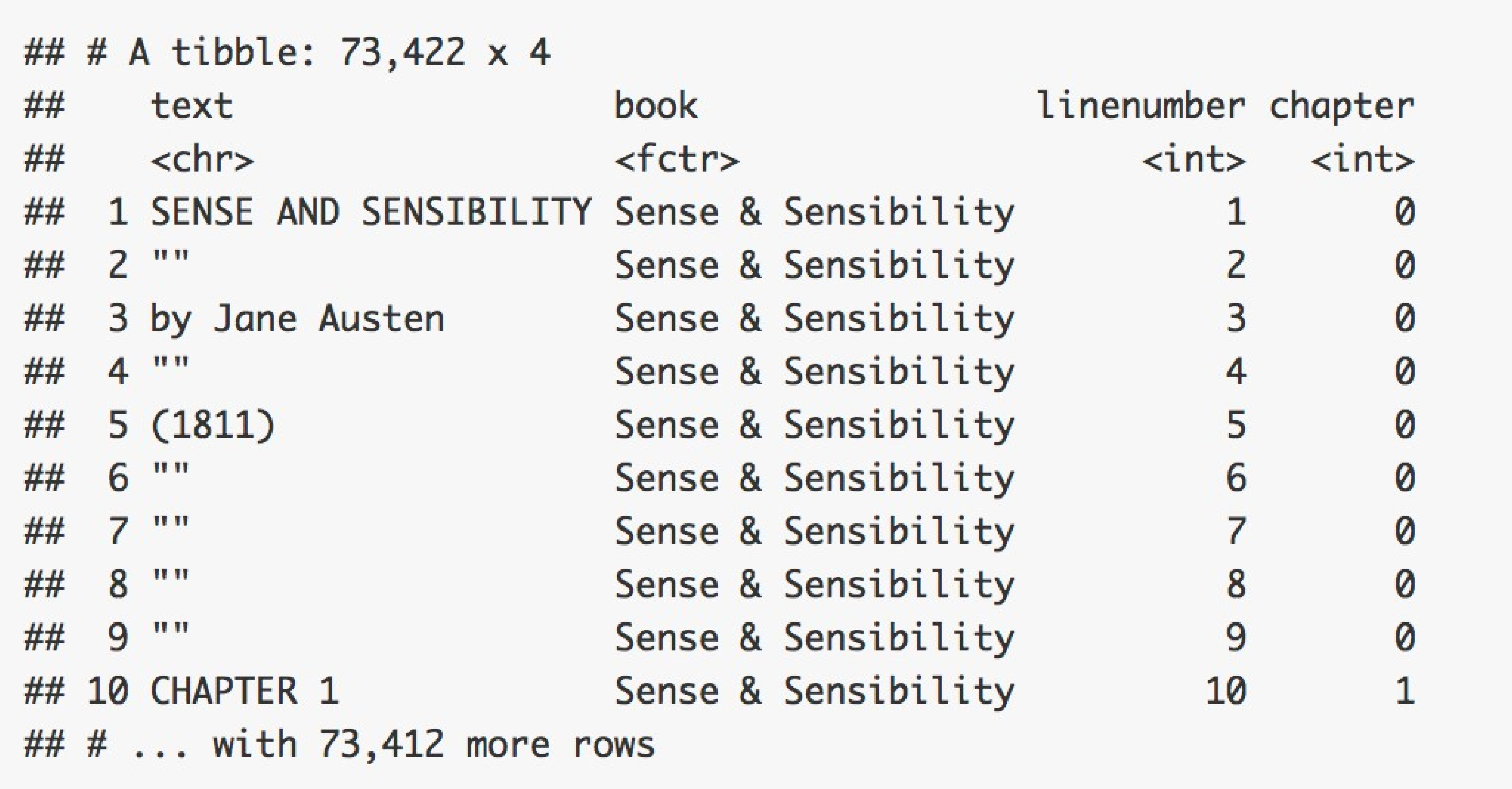
Setup

```
original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
   chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]", ignore_case = TRUE)))) %>%
  ungroup()


View(original_books) # words from all novels
```

Chapter Words

- The words in the order that they appear in the text.
- Note the first line is the title of the book.



```
## # A tibble: 73,422 x 4
##    text                  book                linenumber chapter
##    <chr>                 <fctr>                   <int>   <int>
##  1 SENSE AND SENSIBILITY Sense & Sensibility          1       0
##  2 ""                    Sense & Sensibility          2       0
##  3 by Jane Austen        Sense & Sensibility          3       0
##  4 ""                    Sense & Sensibility          4       0
##  5 (1811)                Sense & Sensibility          5       0
##  6 ""                    Sense & Sensibility          6       0
##  7 ""                    Sense & Sensibility          7       0
##  8 ""                    Sense & Sensibility          8       0
##  9 ""                    Sense & Sensibility          9       0
## 10 CHAPTER 1             Sense & Sensibility         10       1
## # ... with 73,412 more rows
```

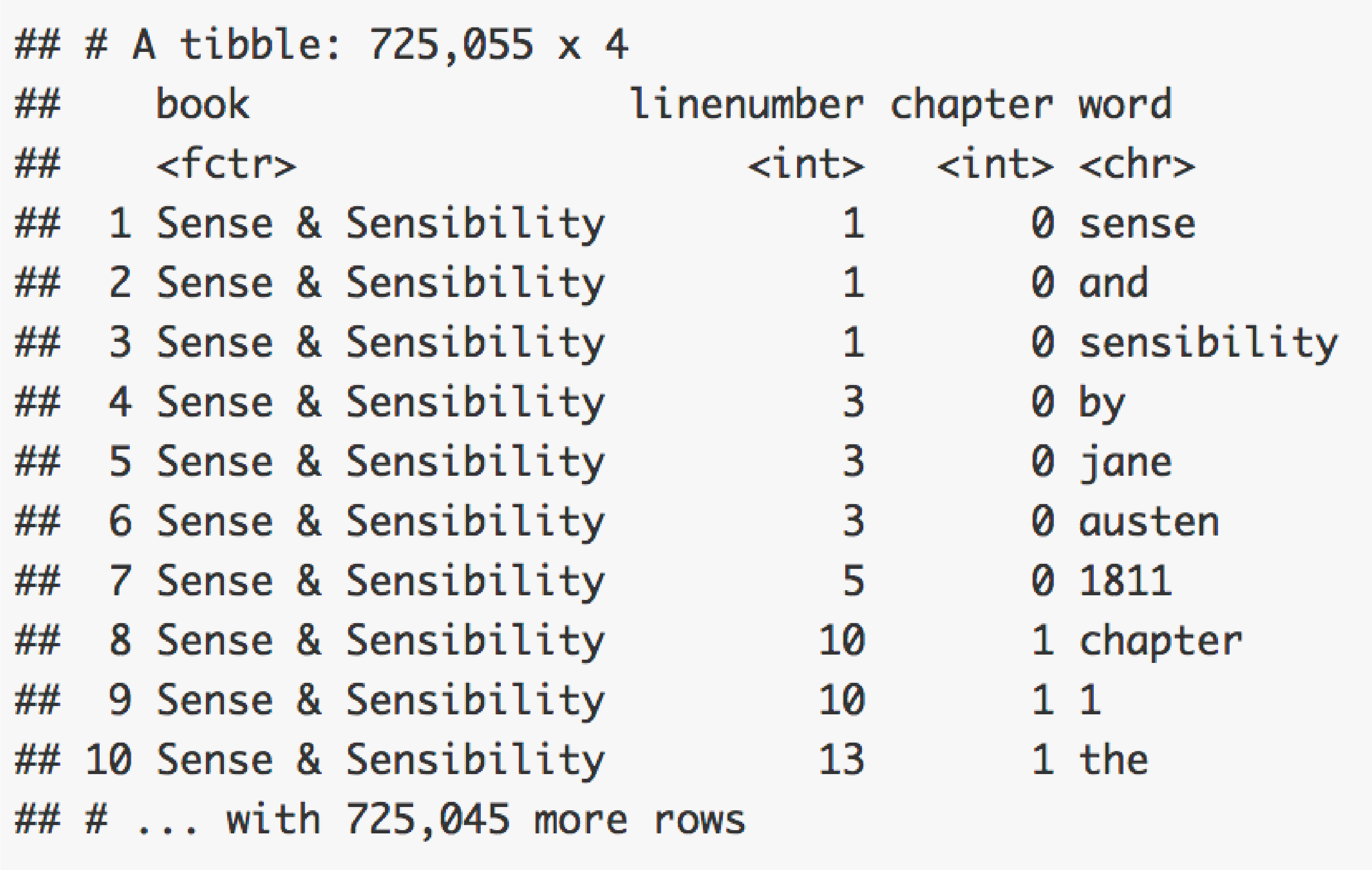
Unnesting Book Words

We need the words in list (un-nested) to work with them.

```
tidy_books <- original_books %>%
  unnest_tokens(word, text) #make a list of words from the paragraphs


View(tidy_books)
```

Unnested Words



```
## # A tibble: 725,055 x 4
##    book                linenumber chapter word
##    <fctr>                   <int>   <int> <chr>
##  1 Sense & Sensibility          1       0 sense
##  2 Sense & Sensibility          1       0 and
##  3 Sense & Sensibility          1       0 sensibility
##  4 Sense & Sensibility          3       0 by
##  5 Sense & Sensibility          3       0 jane
##  6 Sense & Sensibility          3       0 austen
##  7 Sense & Sensibility          5       0 1811
##  8 Sense & Sensibility         10       1 chapter
##  9 Sense & Sensibility         10       1 1
## 10 Sense & Sensibility         13       1 the
## # ... with 725,045 more rows
```

When words are in one-word-per-row format, manipulation with tidy tools like *dplyr* is possible

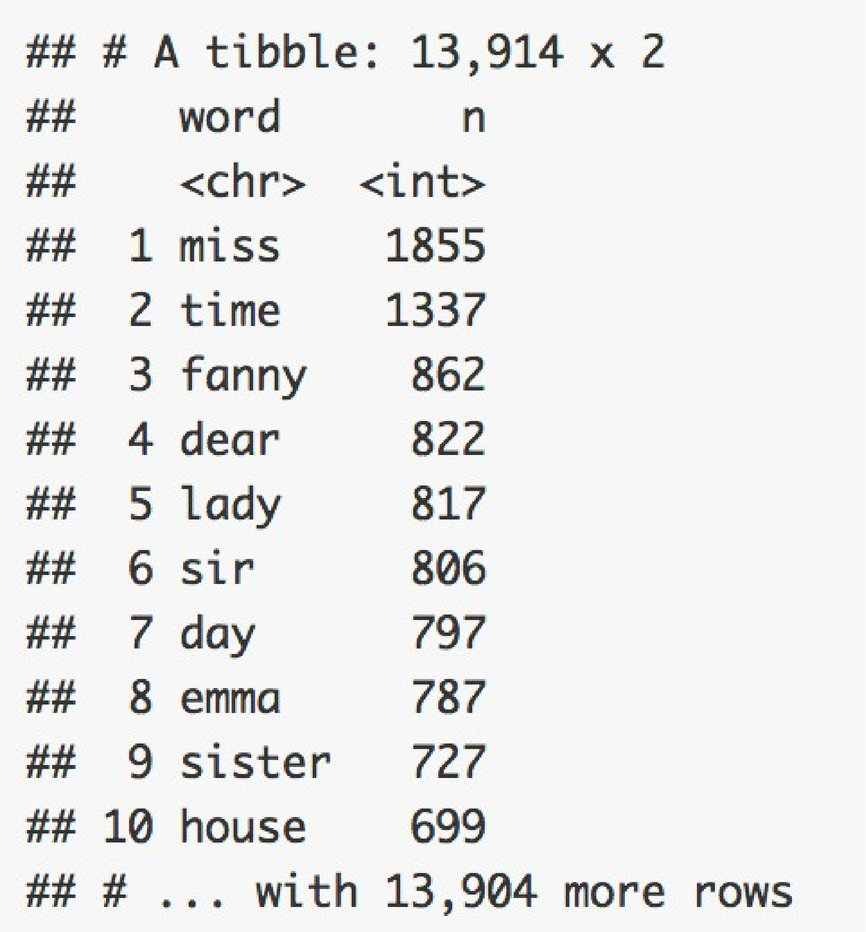
Stop Words

- Remove *stop words:* words which do not add any distinguishing information to a body of text.
  - Contractions: hasn’t, didn’t won’t
  - In-betweens: been, is, had, having

```
data("stop_words")
View(stop_words)
cleaned_books <- tidy_books %>% anti_join(stop_words)
# anti_join() returns all rows from x where there are not
matching values in y, keeping just columns from x.
```

Counting Common Words Across All Books

```
cleaned_books %>%
  count(word, sort = TRUE)
```



```
## # A tibble: 13,914 x 2
##      word       n
##     <chr>   <int>
##   1 miss     1855
##   2 time     1337
##   3 fanny     862
##   4 dear      822
##   5 lady      817
##   6 sir       806
##   7 day       797
##   8 emma      787
##   9 sister    727
## 10 house     699
## # ... with 13,904 more rows
```

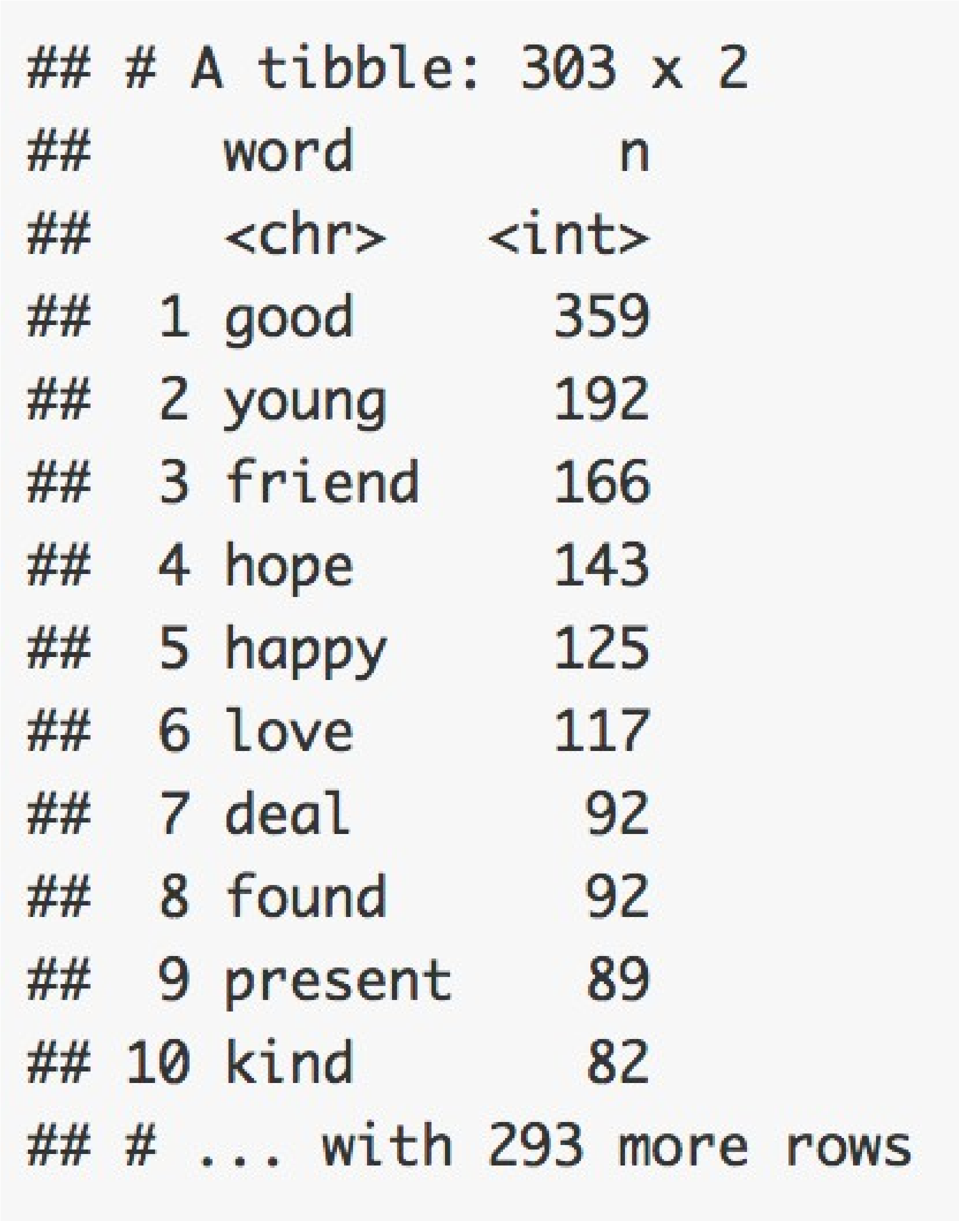
Joy in Emma

- We will consider the common words having scores indicating that they are of Joy, according to the nrc lexicon in the novel, Emma

```
nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
tidy_books %>%
  filter(book == "Emma") %>%
  semi_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

Oh Joy ...

```
tidy_books %>%
  filter(book == "Emma") %>%
  semi_join(nrcjoy) %>%
 count(word, sort = TRUE)
```



```
## # A tibble: 303 x 2
##     word        n
##     <chr>   <int>
##   1 good      359
##   2 young     192
##   3 friend    166
##   4 hope      143
##   5 happy     125
##   6 love      117
##   7 deal       92
##   8 found      92
##   9 present    89
## 10 kind       82
## # ... with 293 more rows
```

We find counts of the *joy* words in the novel, Emma

How Does Sentiment Change?
(In each novel?)

```
library(tidyr)

bing <- get_sentiments("bing")



janeaustensentiment <- tidy_books %>%

  inner_join(bing) %>%

  count(book, index = linenumber %/% 80, sentiment)
        %>% spread(sentiment, n, fill = 0) %>%
     mutate(sentiment = positive - negative)
```

What Are The Most Common Good and Bad Words?

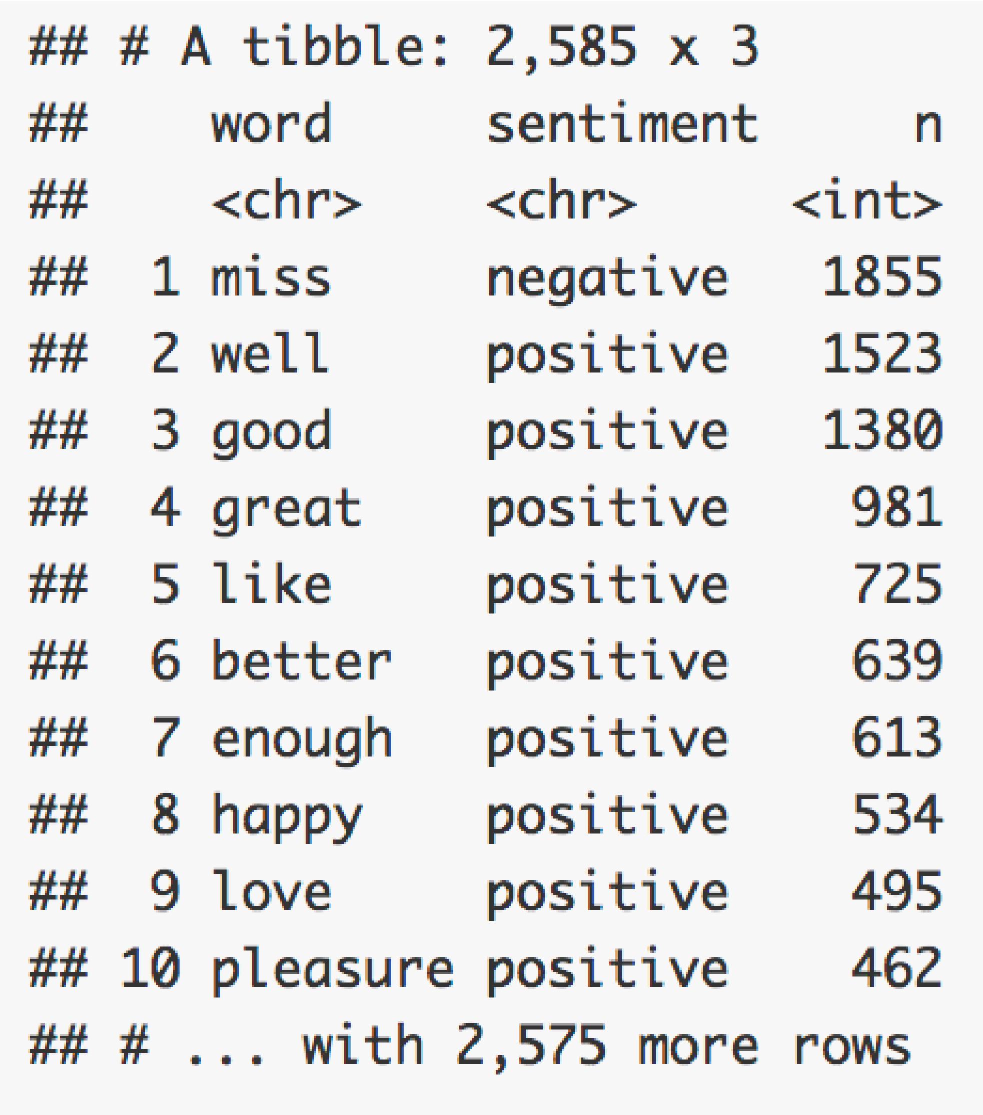
- Count the common positive words across the books.

```
bing_word_counts <- tidy_books %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()


View(bing_word_counts)
```

Such Positivity ...

View(bing_word_counts)

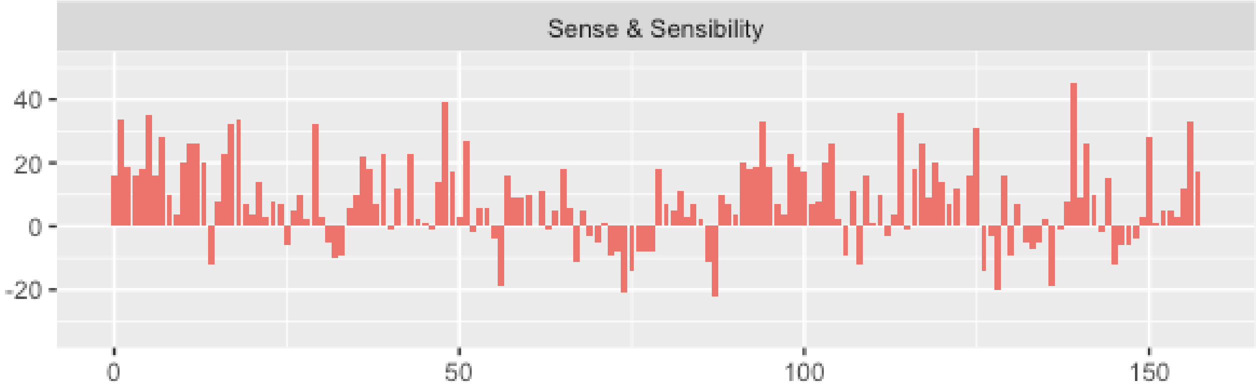


```
## # A tibble: 2,585 x 3
##        word sentiment     n
##       <chr>     <chr> <int>
##  1 miss     negative   1855
##  2 well     positive   1523
##  3 good     positive   1380
##  4 great    positive    981
##  5 like     positive    725
##  6 better   positive    639
##  7 enough   positive    613
##  8 happy    positive    534
##  9 love     positive    495
## 10 pleasure positive    462
## # ... with 2,575 more rows
```

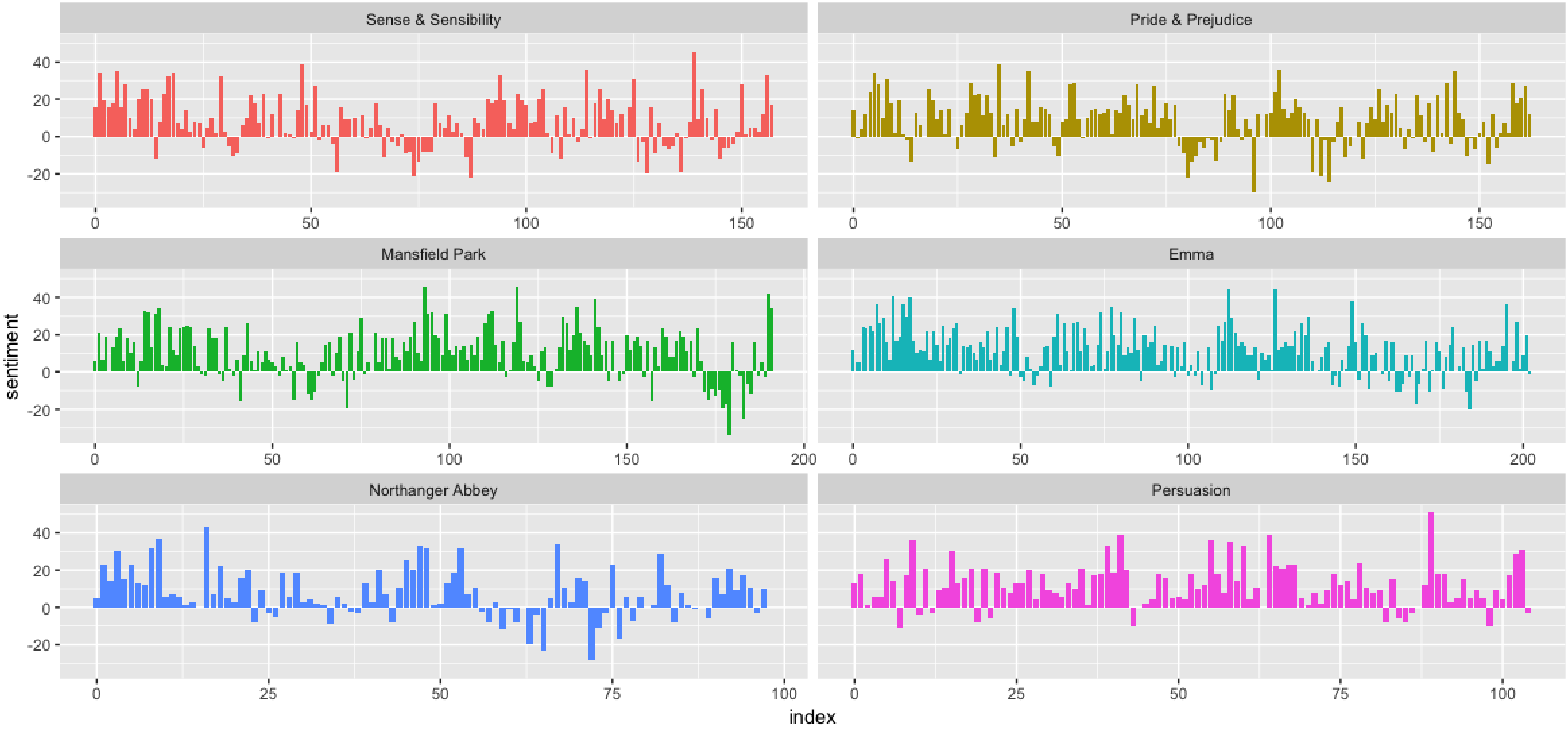
Plot the Good and Bad Words Across Each Book

- # plot the sentiment in each book

```
ggplot(janeaustensentiment, aes(index,
sentiment, fill = book)) + geom_bar(stat =
"identity", show.legend = FALSE) +
facet_wrap(~book, ncol = 2, scales = "free_x")
```



Plot the Good and Bad Words Across Each Book

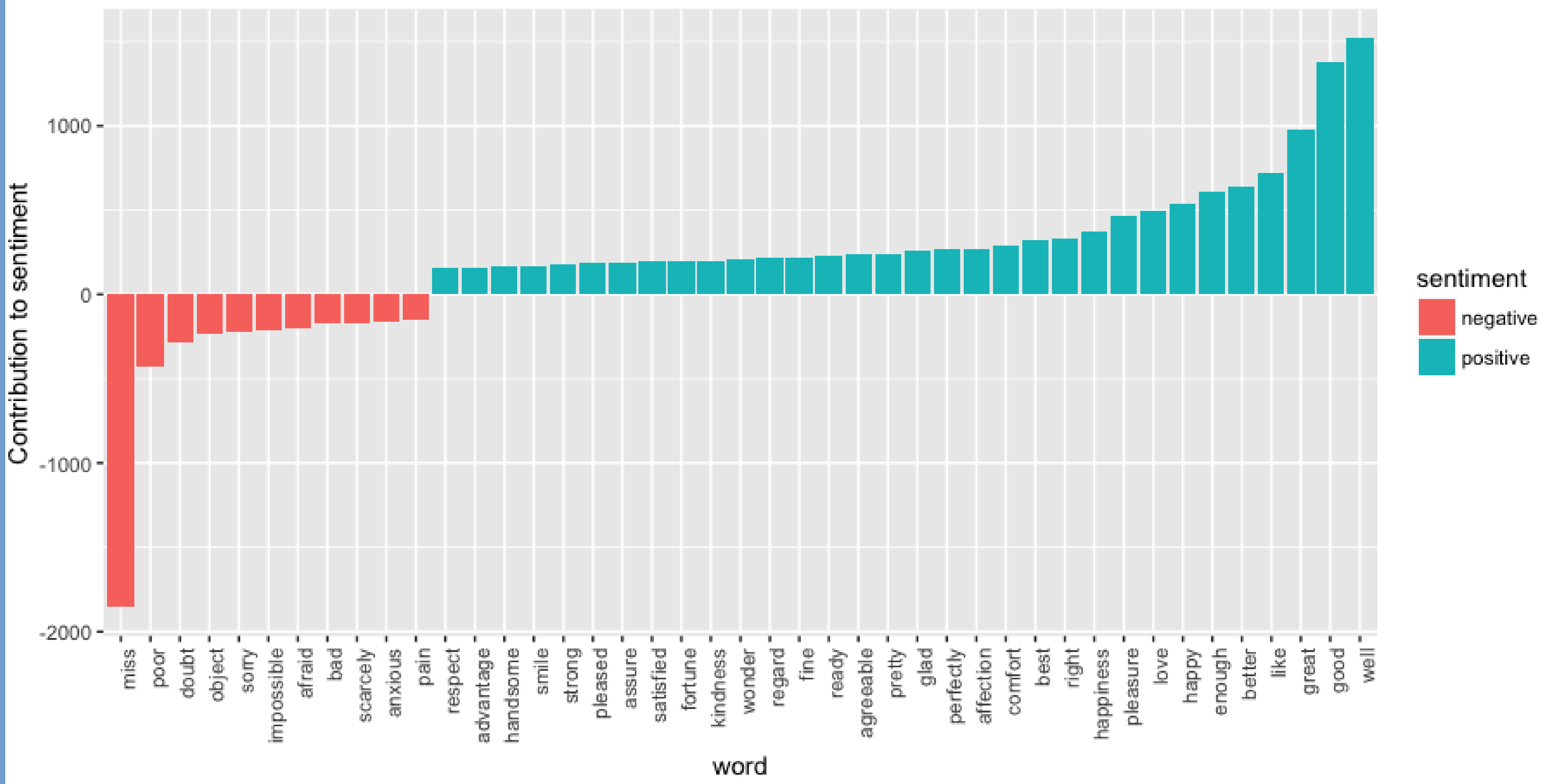


Plot of The Common
Positive and Negative Words

- Plot the common positive words across the books.

```
bing_word_counts %>%
  filter(n > 150) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
 ylab("Contribution to sentiment")
```

Plot of Positive and Negative Sentiment Words


Contribution to sentiment
miss
poor
doubt
object
sorry
impossible
afraid
bad
scarcely
anxious
pain
respect
advantage
handsome
smile
strong
pleased
assure
satisfied
fortune
kindness
wonder
regard
fine
ready
agreeable
pretty
glad
perfectly
affection
comfort
best
right
happiness
pleasure
love
happy
enough
better
like
great
good
well
word
sentiment
🟥 negative
🟩 positive