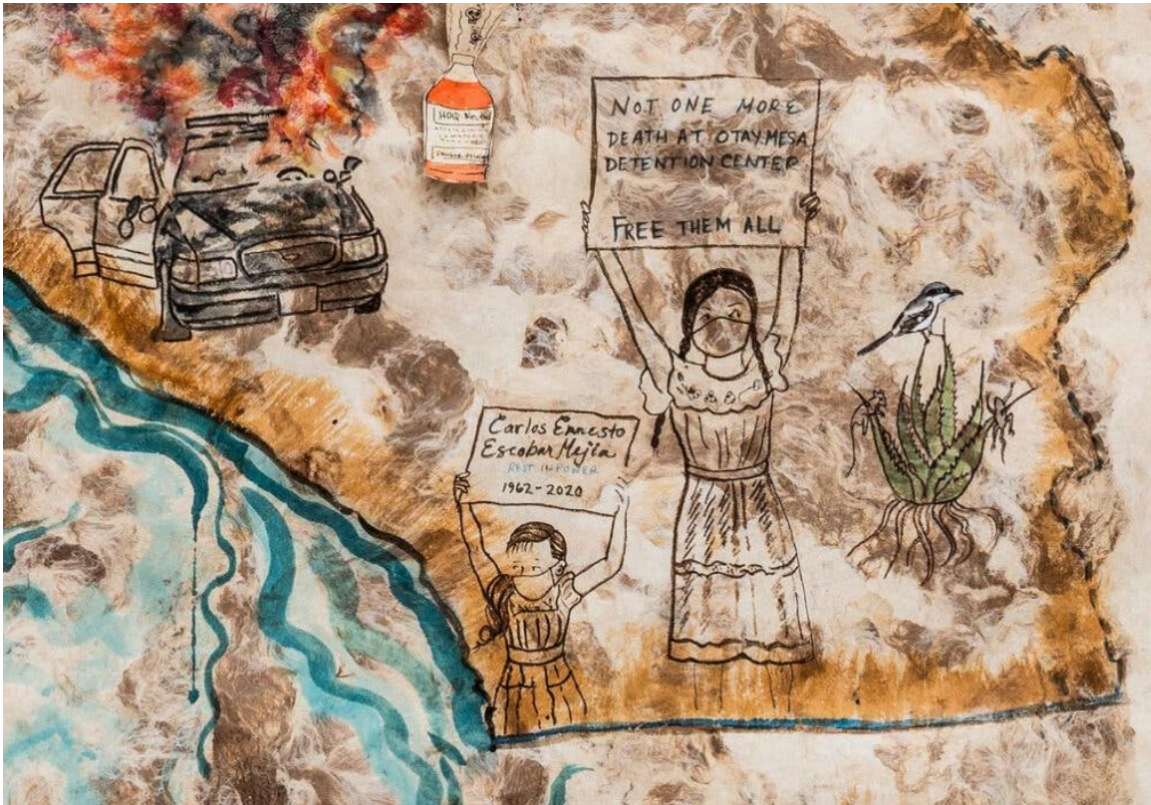


# TEXT MINING WITH R

---

DR. ALVIN ANG



# CONTENTS

<b>I. Step 1: Install Tidyverse</b> .....	<b>3</b>
<b>A. Installing Tidyverse into Linux Mint</b> .....	<b>3</b>
<b>II. Step 2: Load Packages</b> .....	<b>4</b>
<b>III. Step 3: Read in the Text</b> .....	<b>5</b>
<b>IV. Step 4: Tibble</b> .....	<b>6</b>
<b>V. Step 5: Tokenization</b> .....	<b>7</b>
<b>VI. Step 6: Removing Stop Words</b> .....	<b>8</b>
<b>A. Showing the List of Stop Words</b> .....	<b>9</b>
<b>VII. Step 7: Count the Most Frequent Words</b> .....	<b>11</b>
<b>VIII. Step 8: Plotting the Most Frequent Words</b> .....	<b>12</b>
<b>IX. Step 9: Create a Word Cloud</b> .....	<b>14</b>
<b>X. Step 10: Analyze Text Sentiments</b> .....	<b>15</b>
<b>A. NRC</b> .....	<b>15</b>
1. Get Sentiments .....	15
2. Filter Sentiments.....	16
3. Inner Join NRC “Fear” to Text File + Do Word Count .....	16
4. Visualize .....	17
<b>B. Bing</b> .....	<b>18</b>
1. Get Sentiments .....	18
2. Inner Join Sentiments to Text File.....	19
3. Do Word Count + Inner Join Sentiments to Text File .....	20
4. Visualize .....	21
<b>C. AFINN</b> .....	<b>22</b>
1. Get Sentiments .....	22
2. Inner Join Sentiments to Text File.....	22
3. Do Word Count + Inner Join Sentiments to Text File .....	23
4. Visualize .....	23
<b>About Dr. Alvin Ang</b> .....	<b>24</b>

---

## I. STEP 1: INSTALL TIDYVERSE

---

File: <https://www.alvinang.sg/s/Text-Mining-with-R-by-Dr-Alvin-Ang.R>

```
#Step 0: Getting the Files  
#https://www.alvinang.sg/s/eisenhower.txt
```

```
#Step 1: Installing Tidyverse into R  
installed.packages("tidyverse", dependencies = TRUE)  
"
```

- <https://www.tidyverse.org/>

### A. INSTALLING TIDYVERSE INTO LINUX MINT

- You most probably have no issues installing Tidyverse into R using Windows.
- But Linux Mint is tough.
- Do the following:
  - `sudo apt install g++`
  - `sudo apt-get update`
  - `sudo apt-get install libcurl4-openssl-dev`
  - `sudo apt-get install r-base-dev.`
  - reboot your laptop
  - reinstall tidyverse:
    - `install.packages("tidyverse", dependencies=TRUE)`
  - `sudo apt install libssl-dev libxml2-dev`

---

## II. STEP 2: LOAD PACKAGES

---

```
#Step 2: Load Packages  
library(tidyverse)  
library(tibble)  
library(tidyr)  
library(dplyr)  
library(readxl)  
library(ggplot2)  
library(ggthemes)  
library(lubridate)  
library(tidytext)  
library(wordcloud2)  
library(stringr)  
library(textdata)  
#
```

---

### III. STEP 3: READ IN THE TEXT

---

File can be found here: <https://www.alvinang.sg/s/eisenhower.txt>

```
#Step 3: Import Text File  
text <- readLines(file.choose())  
  
text
```

The screenshot displays the R Studio interface. The top-left pane shows the source editor with the following R code:

```
4 library(dplyr)  
5 library(readxl)  
6 library(ggplot2)  
7 library(lubridate)  
8 library(tidytext)  
9 library(wordcloud2)  
10 library(stringr)  
11 library(textdata)  
12  
13 text <- readLines(file.choose())  
14 text  
15
```

The top-right pane shows the Environment window with a variable named 'text' of type 'chr [1:4]' containing the text: "Dwight David \"Ike\" Eisenhower (/ 'aIz...".

The bottom-left pane shows the Console window with the following output:

```
> text <- readLines(file.choose())  
> text  
[1] "Dwight David \"Ike\" Eisenhower (/ 'aIzənhəʊ.ər/ EYE-zən-how-ər; October  
14, 1890 - March 28, 1969) was an American politician and Army general who ser  
ved as the 34th President of the United States from 1953 until 1961. He was a f  
ive-star general in the United States Army during World War II and served as Su  
preme Commander of the Allied Expeditionary Forces in Europe. He was responsibl  
e for planning and supervising the invasion of North Africa in Operation Torch  
in 1942-43 and the successful invasion of France and Germany in 1944-45 from t  
he Western Front."
```



```
#Step 4: Tibble it (Convert to Dataframe)
length = length(text)
tb <- tibble(line = 1:length, text = text)

tb
```

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for Step 4. Lines 32-35 are highlighted with a red box. The code is:
 

```
length = length(text)
tb <- tibble(line = 1:length, text = text)
tb
```
- Environment Pane:** Shows the variable 'tb' as a tibble with 4 observations and 2 variables. The 'length' variable is shown as 4L, and the 'text' variable is shown as chr [1:4] "Dwight David \"Ike...".
- Console:** Shows the output of the tibble 'tb' as a 4x2 data frame. The first four rows are highlighted with a red box:
 

```
1 1 "Dwight David \"Ike\" Eisenhower (/ 'aɪzənhaʊ.ər/ EYE-zən-how-ər; Octo...
2 2 "Eisenhower was born in Denison, Texas, and raised in Kansas in a lar...
3 3 "Eisenhower entered the 1952 presidential race as a Republican, in or...
4 4 "On the domestic front, Eisenhower was a moderate conservative who co...
```
- Annotations:** Red text in the bottom right corner explains the output:
  - Each paragraph is considered as 1L
  - 4 Paragraphs = 4L
  - 4L of texts are stored into the dataframe "tb" by the function "tibble"
  - length(text) takes the full length of the text and stores it into "length"

```
#Step 5: Unnest (Tokenization)
tb_un <- tb %>%
  unnest_tokens(word, text)

tb_un
```

The screenshot shows the R Studio interface. The script editor on the left contains the following code:

```
38 #Step 5: Unnest (Tokenization)
39 tb_un <- tb %>%
40   unnest_tokens(word, text)
41
42 tb_un
43 #-----
44
```

The Environment pane on the right shows the data frame `tb_un` with 705 observations and 2 variables, highlighted with a red box. The Values pane shows the structure of the data:

```
length 4L
text chr [1:4] "Dwight David \\'Ik..."
```

The Console pane shows the output of the `tb_un` command:

```
# A tibble: 705 × 2
  line word
  <int> <chr>
1     1 dwight
2     1 david
3     1 ike
4     1 eisenhower
5     1 'aIzənhəʊ.ər
6     1 eye
7     1 zən
8     1 how
9     1 ər
```

The output is highlighted with a red box. To the right of the console output, there is a red text annotation:

tokenization is the process of separating each individual word into tokens  
i.e. 1 token = 1 word

## Stop Words

These words include:

- a
- of
- on
- I
- for
- with
- the
- at
- from
- in
- to

Stop words are useless words when it comes to text analysis, because they don't have meaning. We will remove them.

```
#Step 6: Remove Stop words  
sw = stop_words  
  
tb_un_rm <- tb_un %>%  
  anti_join(stop_words)  
  
tb_un_rm
```



#### A. SHOWING THE LIST OF STOP WORDS

The screenshot shows the R Studio interface. The source editor on the left contains R code for text mining. Line 46, `sw = stop_words`, is highlighted with a red box. A red arrow points from this box to the Environment pane on the right, where the variable `sw` is listed with 1149 observations and 2 variables. The console at the bottom shows the command `> View(sw)` being executed.

```
41
42 tb_un
43 #-----
44
45 #Step 6: Remove Stop words
46 sw = stop_words
47
48 tb_un_rm <- tb_un %>%
49   anti_join(stop_words)
50
51 tb_un_rm
52 #-----
53
54 #Step 7: Count the Most Frequent Words
55 tb_un_rm_c = tb_un_rm %>%
56
```

Environment | History | Connections | Tutorial

Global Environment

Data

sw	1149 obs. of 2 variables
tb	4 obs. of 2 variables
tb_un	705 obs. of 2 variables
tb_un_rm	388 obs. of 2 variables

Values

length	4L
--------	----

Console

```
R 3.6.3 ~/
# ... with 378 more rows
> View(sw)
```

Text Mining with R by Dr. Alvin An... sw

	word	lexicon
1	a	SMART
2	a's	SMART
3	able	SMART
4	about	SMART
5	above	SMART
6	according	SMART
7	accordingly	SMART
8	across	SMART
9	actually	SMART
10	after	SMART
11	afterwards	SMART
12	again	SMART

Environment  
R | Glo  
Data  
sw  
tb  
tb\_un  
tb\_un\_rm  
Values  
length  
Files Plots

these are the list of STOP WORDS

```
> tb_un_rm
# A tibble: 388 x 2
  line word
  <int> <chr>
1     1 dwight
2     1 david
3     1 ike
4     1 eisenhower
5     1 'aɪzənhɑː.ər
6     1 eye
7     1 zən
8     1 əɾ
9     1 october
10    1 14
```

stop words have been removed

```
#Step 7: Count the Most Frequent Words
tb_un_rm_c = tb_un_rm %>%
  count(word, sort = TRUE)

tb_un_rm_c
```

```
> tidy_df_rm %>%
+   count(word, sort = TRUE)
# A tibble: 300 × 2
   word          n
  <chr>      <int>
1 eisenhower    12
2 war           7
3 army          5
4 served        5
5 invasion      4
6 military      4
7 china         3
8 french        3
9 nuclear       3
10 soviet        3
# ... with 290 more rows
>
```

```
#Step 8: Plotting the Most Frequent Words
tb_un_rm_c_plt = tb_un_rm_c %>%
  filter(n>2) %>%
  mutate(word = reorder(word, n)) %>%

  ggplot(aes(word, n)) + geom_col(fill = "darkred") + theme_fivethirtyeight() +
  xlab(NULL) + ylab("Word Count") + coord_flip() + ggtitle("Word Usage in Eisenhower.txt")

tb_un_rm_c_plt
```

	word	n
1	eisenhower	12
2	war	7
3	army	5
4	served	5
5	invasion	4
6	military	4
7	china	3
8	french	3
9	nuclear	3
10	soviet	3
11	strong	3

n is the number of times the word appears

we need this column to be named "n"

to be used in the later code

...

Install Packages

Install from: Repository (CRAN)

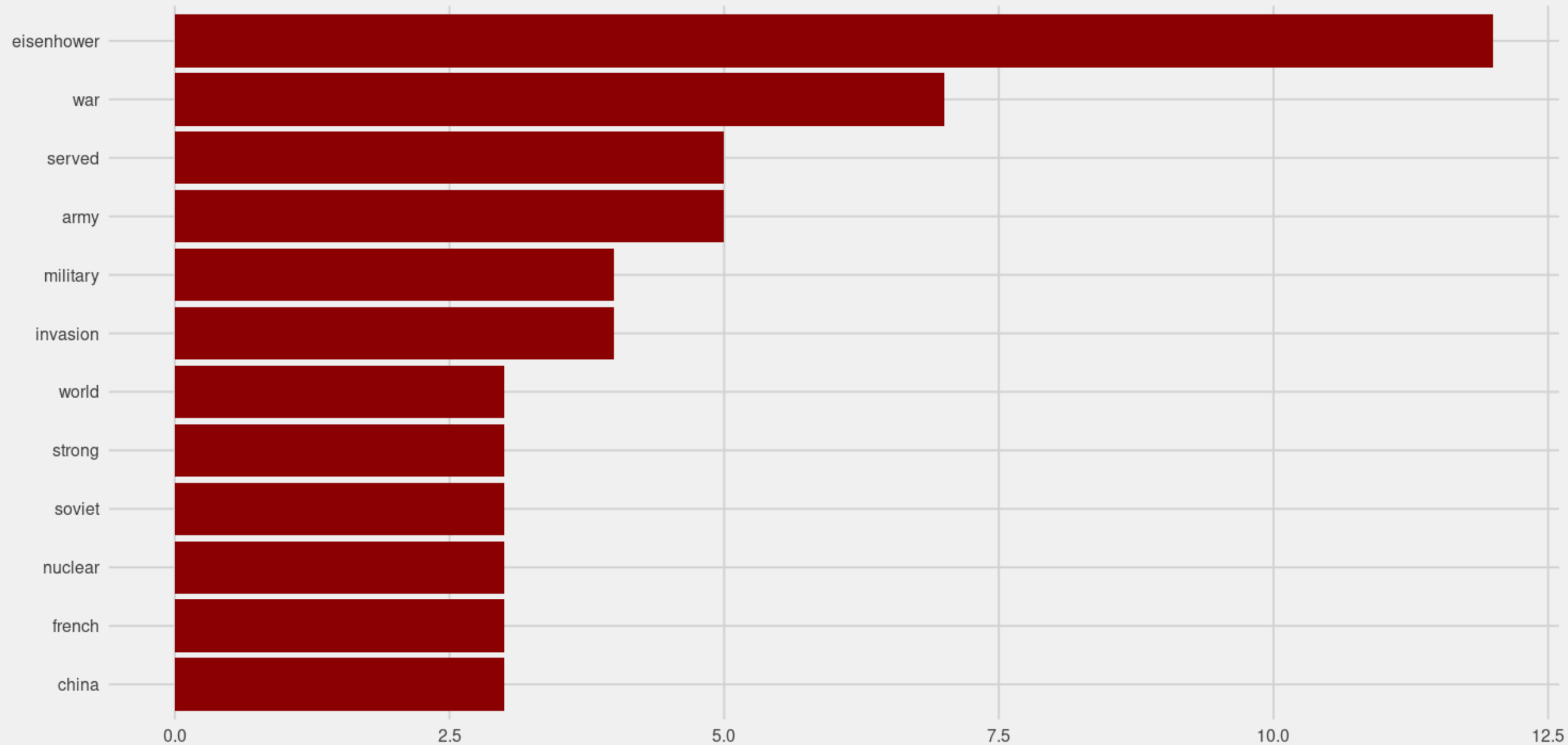
Packages (separate multiple with space or comma): ggthemes

Install to Library: /home/dralvin/R/x86\_64-pc-linux-gnu-library/3.6 [Default]

Install dependencies

Install Cancel

## Word Usage in Eisenhower.txt







- Sentiment lexicons give emotions to a given text.
- There are three lexicons that can be used:

A. NRC

1. GET SENTIMENTS

```
#Step 10: Analyze Text Sentiments
#10a) NRC
# nrc categorizes words as POSITIVE / NEGATIVE / ANGER / ANTICIPATION / DISGUST
# FEAR / JOY / SADNESS / SURPRISE AND TRUST

#10a)(i) Get Sentiments
nrc = get_sentiments("nrc")
```

```
Selection: 1 nrc is a big library... u have to wait long time to load it...
trying URL 'http://saifmohammad.com/WebDocs/NRC-Emotion-Lexicon.zip'
Content type 'application/zip' length 24436570 bytes (23.3 MB)
=====
```

	word	sentiment
15	abduct	trust
16	abduction	fear
17	abduction	negative
18	abduction	sadness
19	abduction	surprise
20	aberrant	negative
21	aberration	disgust
22	aberration	negative
23	abhor	anger
24	abhor	disgust
25	abhor	fear
26	abhor	negative
27	abhorrent	anger
28	abhorrent	disgust

nrc labels different sentiments to different words...

## 2. FILTER SENTIMENTS

```
#10a)(ii) Filter Sentiments
nrc_sentiment <- get_sentiments("nrc") %>%
  filter(sentiment == "fear")
```

## 3. INNER JOIN NRC "FEAR" TO TEXT FILE + DO WORD COUNT

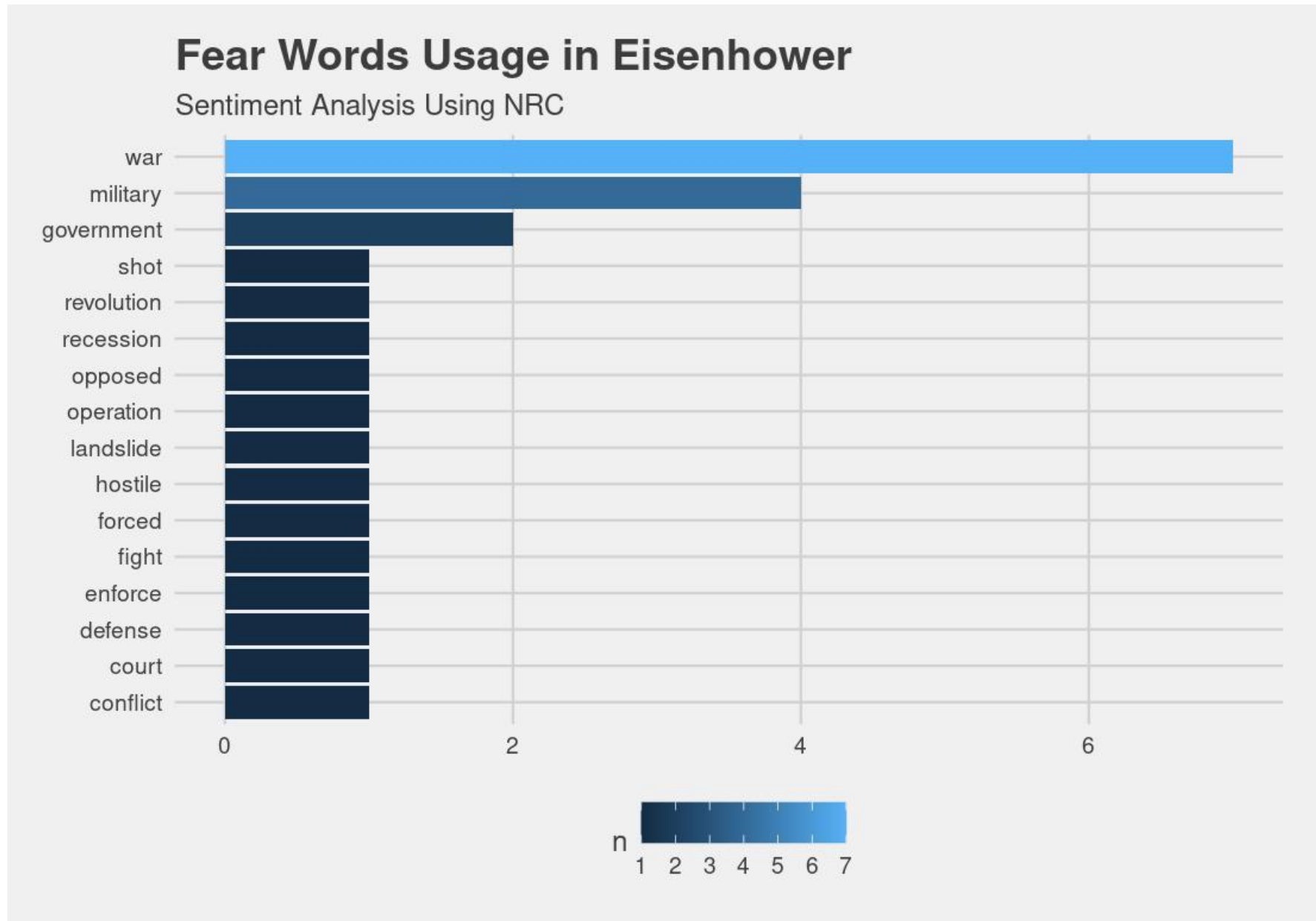
```
#10a)(iii) Inner Join Sentiments to Text File + Do Word Count
tb_un_rm_nrc = tb_un_rm %>%
  inner_join(nrc_sentiment) %>%
  count(word, sort = TRUE)

tb_un_rm_nrc
```

```
word      n
<chr>    <int>
1 war      7
2 military 4
3 government 2
4 conflict 1
5 court    1
6 defense  1
7 enforce  1
8 fight    1
9 forced   1
10 hostile 1
11 landslide 1
12 operation 1
13 opposed 1
14 recession 1
15 revolution 1
16 shot    1
>
```

From the Eisenhower.txt, we see that 16 words affiliate with "fear", the top word striking fear is "war".

```
#10a)(iv) Visualize
tb_un_rm_nrc %>%
  filter(n > 0) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill=n)) + theme_fivethirtyeight() + geom_col() +
  xlab(NULL) + coord_flip() + ylab("Word Count") +
  ggtitle("Fear Words Usage in Eisenhower",
          subtitle = "Sentiment Analysis Using NRC")
```

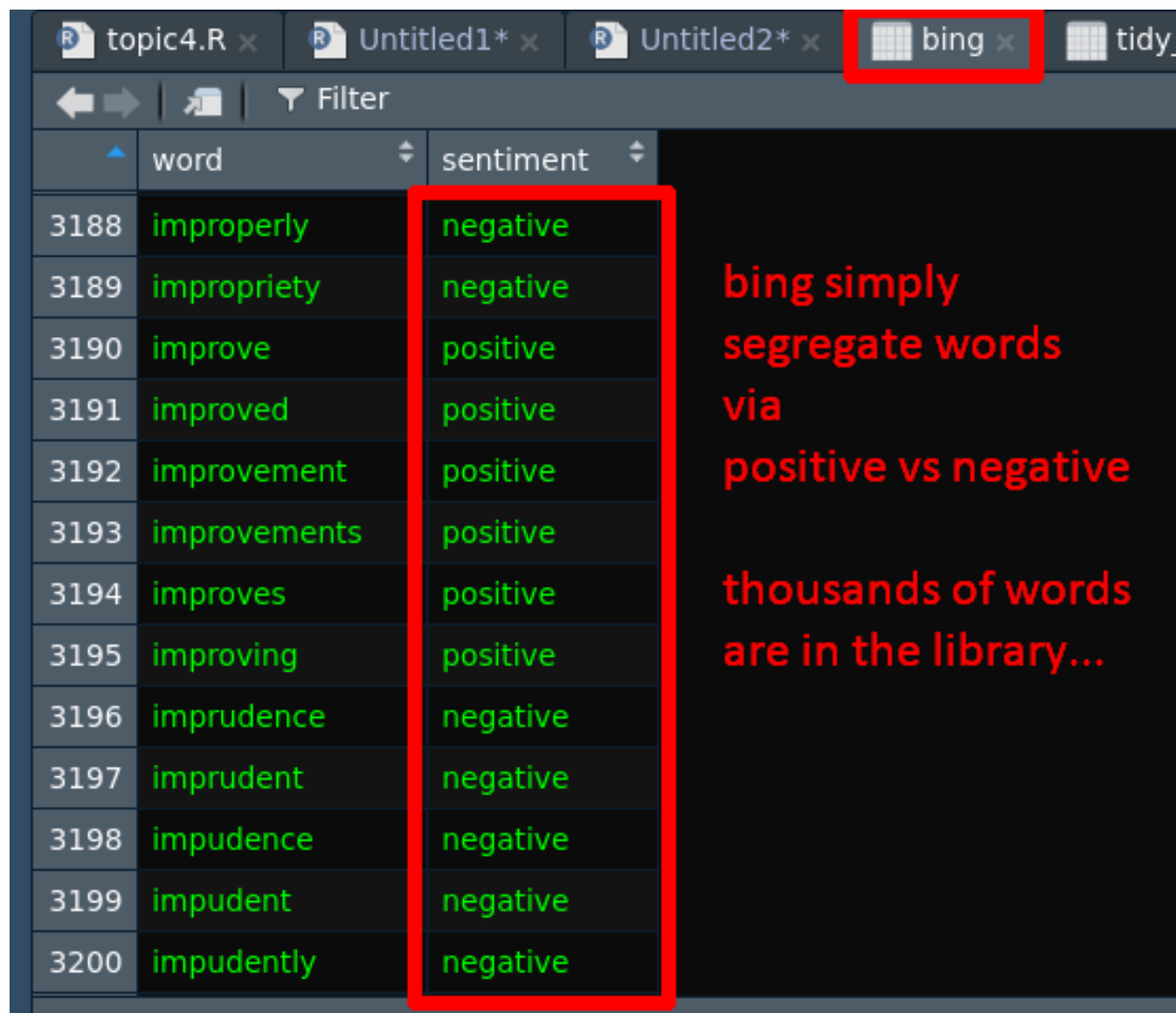




## B. BING

### 1. GET SENTIMENTS

```
#10b) Bing  
# Bing categorizes words as Positive or Negative  
  
#10b)(i) Get Sentiments  
bing = get_sentiments('bing')
```



	word	sentiment
3188	improperly	negative
3189	impropriety	negative
3190	improve	positive
3191	improved	positive
3192	improvement	positive
3193	improvements	positive
3194	improves	positive
3195	improving	positive
3196	imprudence	negative
3197	imprudent	negative
3198	impudence	negative
3199	impudent	negative
3200	impudently	negative

**bing simply segregate words via positive vs negative thousands of words are in the library...**



2. INNER JOIN SENTIMENTS TO TEXT FILE

```
#10b)(ii) Inner Join Sentiments to Text File  
tb_un_rm_bing = tb_un_rm %>%  
  inner_join(get_sentiments('bing'))
```



The screenshot shows an RStudio window with a data table titled 'tidy\_df\_rm\_bing'. The table has three columns: 'line', 'word', and 'sentiment'. The data is as follows:

line	word	sentiment
1	supreme	positive
2	successful	positive
3	strong	positive
4	denied	negative
5	tank	negative
6	successful	positive
7	uncomfortable	negative
8	supreme	positive
9	won	positive
10	winner	positive
11	conflict	negative
12	inexpensive	positive
13	expensive	negative
14	won	positive
15	approval	positive
16	strong	positive
17	support	positive
18	supported	positive
19	hostile	negative
20	crisis	negative
21	condemned	negative

All words that have a labeled sentiment are displayed.

3. DO WORD COUNT + INNER JOIN SENTIMENTS TO TEXT FILE

```
#10b)(iii) Do Word Count + Inner Join Sentiments to Text File
tb_un_rm_bing_1 = tb_un_rm_bing %>%
  count(word, sort = TRUE) %>%
  inner_join(tb_un_rm_bing)
```

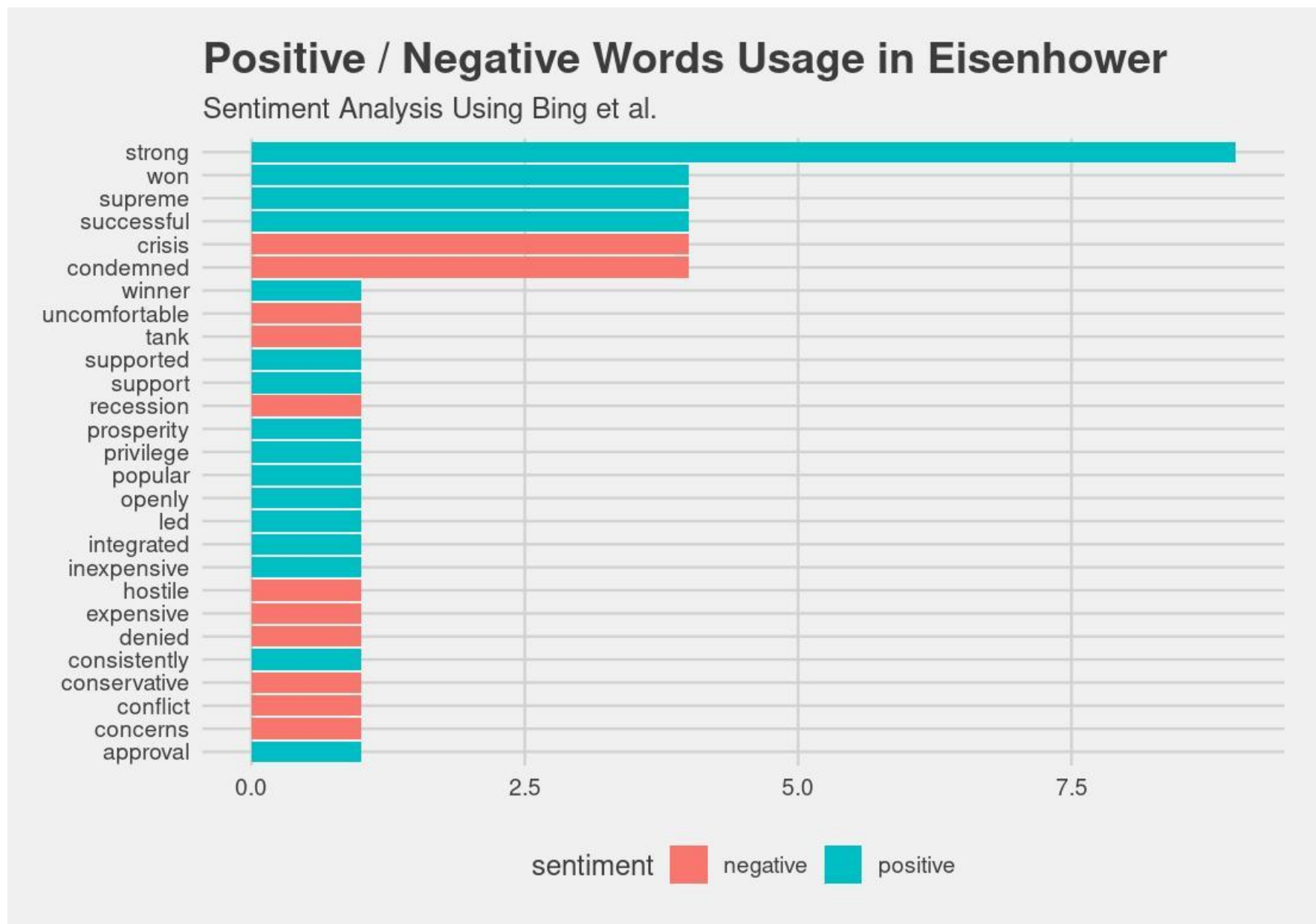


	word	n	line	sentiment
1	strong	3	2	positive
2	strong	3	3	positive
3	strong	3	4	positive
4	condemned	2	3	negative
5	condemned	2	3	negative
6	crisis	2	3	negative
7	crisis	2	3	negative
8	successful	2	1	positive
9	successful	2	2	positive
10	supreme	2	1	positive
11	supreme	2	2	positive
12	won	2	3	positive
13	won	2	3	positive
14	approval	1	3	positive
15	concerns	1	4	negative
16	conflict	1	3	negative
17	conservative	1	4	negative

We created 4 columns (word / n / line / sentiment) by inner joining the “count of words” back to “tb\_un\_rm\_bing”.

We need these columns for plotting in the next section.

```
#10b)(iv) Visualize
tb_un_rm_bing_1 %>%
  filter(n > 0) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) + theme_fivethirtyeight() + geom_col() +
  xlab(NULL) + coord_flip() + ylab("Word Count") +
  ggtitle("Positive / Negative Words Usage in Eisenhower",
          subtitle = "Sentiment Analysis Using Bing et al.")
```

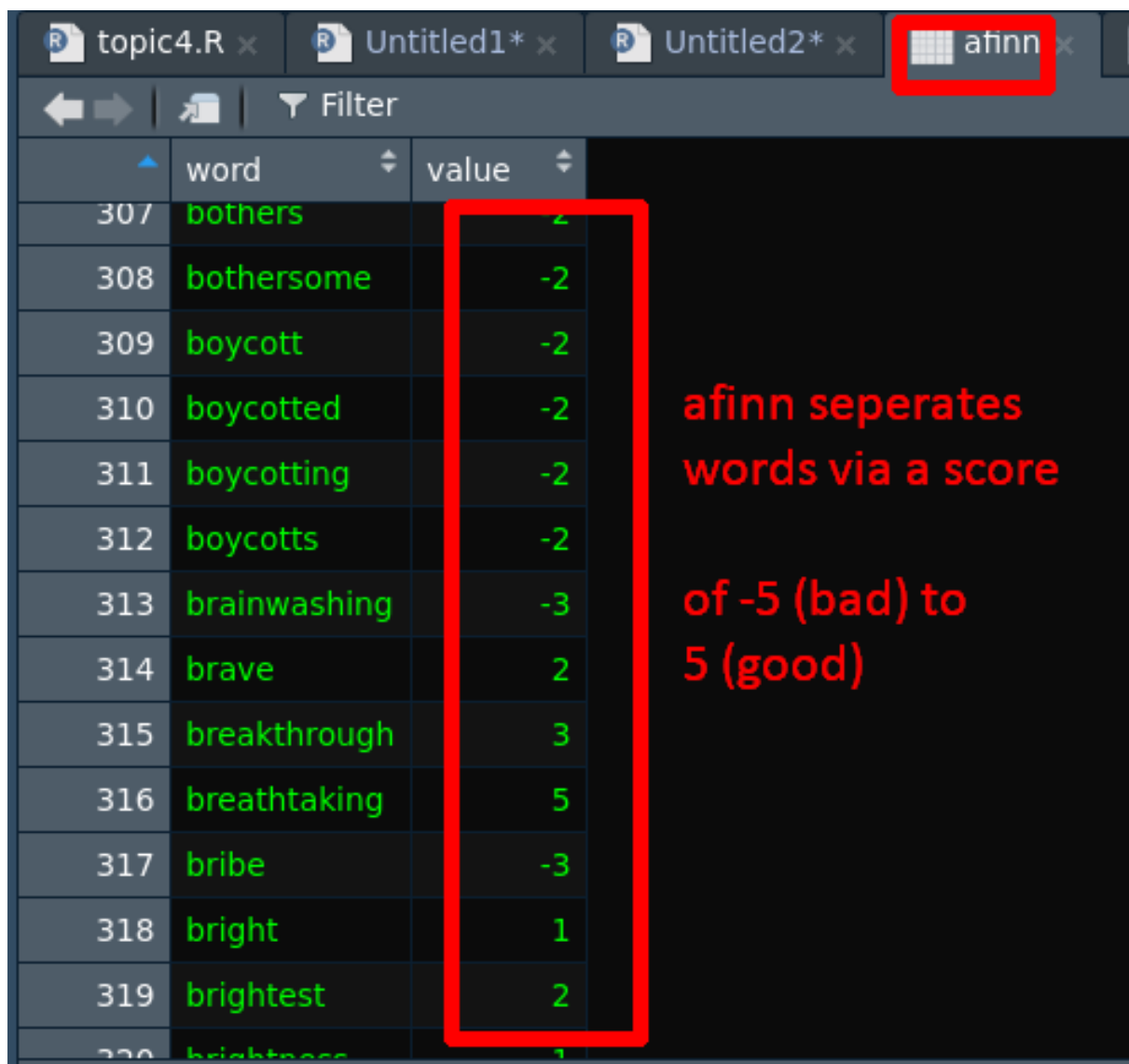


## C. AFINN

### 1. GET SENTIMENTS

```
#10c) AFINN
# AFINN give word scores
# -5 is Negative while 5 is Positive

#10c)(i) Get Sentiments
afinn = get_sentiments("afinn")
```



	word	value
307	bothers	-2
308	bothersome	-2
309	boycott	-2
310	boycotted	-2
311	boycotting	-2
312	boycotts	-2
313	brainwashing	-3
314	brave	2
315	breakthrough	3
316	brehtaking	5
317	bribe	-3
318	bright	1
319	brightest	2
320	brightness	1

afinn separates words via a score of -5 (bad) to 5 (good)

### 2. INNER JOIN SENTIMENTS TO TEXT FILE

```
#10c)(ii) Inner Join Sentiments to Text File
tb_un_rm_afinn = tb_un_rm %>%
  inner_join(get_sentiments('afinn'))
```

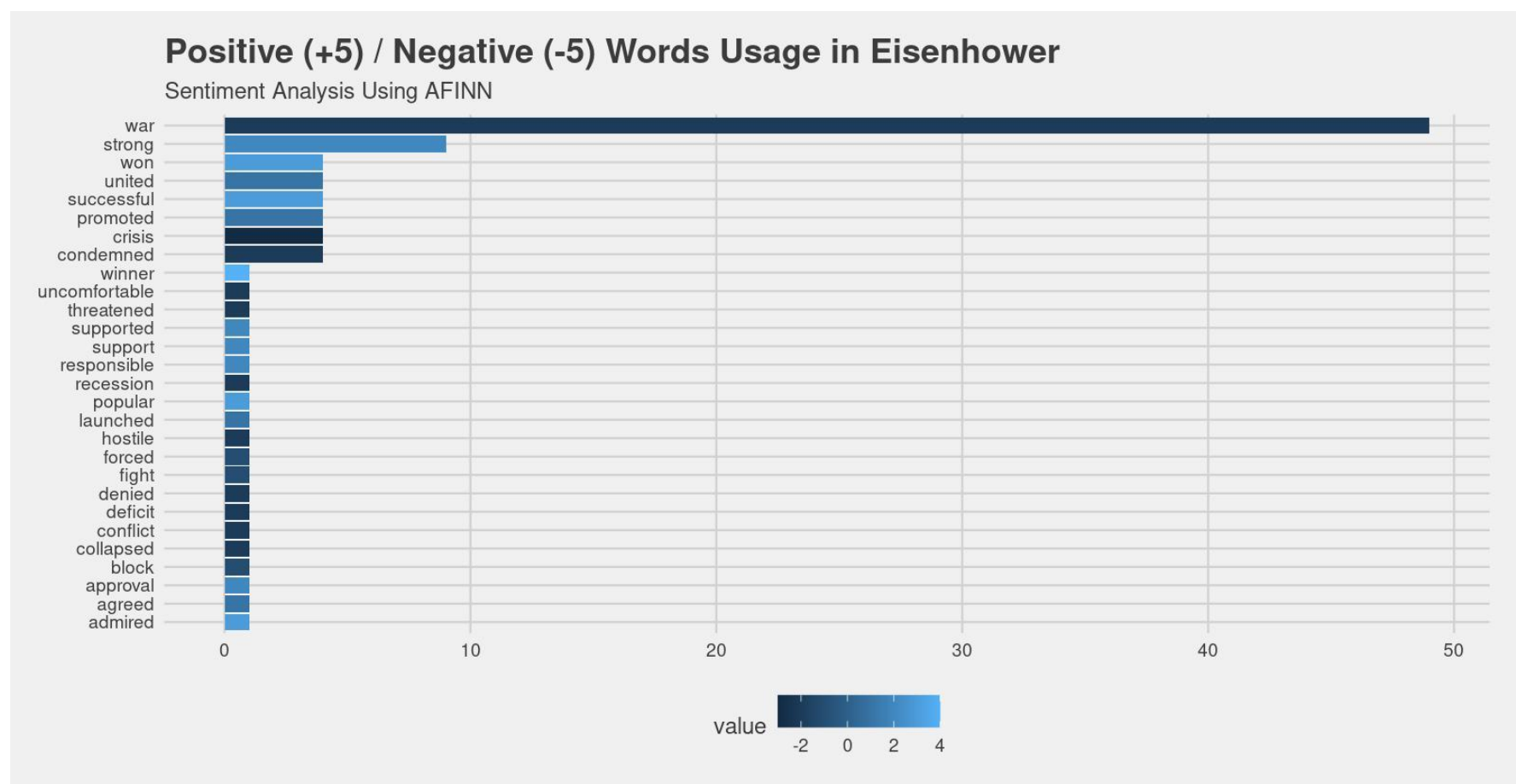


3. DO WORD COUNT + INNER JOIN SENTIMENTS TO TEXT FILE

```
#10c)(iii) Do Word Count + Inner Join Sentiments to Text File
tb_un_rm_afinn_1 = tb_un_rm_afinn %>%
  count(word, sort = TRUE) %>%
  inner_join(tb_un_rm_afinn)
```

4. VISUALIZE

```
#10c)(iv) Visualize
tb_un_rm_afinn_1 %>%
  filter(n > 0) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill=value)) + theme_fivethirtyeight() + geom_col() +
  xlab(NULL) + coord_flip() + ylab("Word Count") +
  ggtitle("Positive (+5) / Negative (-5) Words Usage in Eisenhower",
    subtitle = "Sentiment Analysis Using AFINN")
```





---

## ABOUT DR. ALVIN ANG

---



Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at [www.AlvinAng.sg](http://www.AlvinAng.sg).