Assignment4

Ruining Jia

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(knitr)
library(kableExtra)
```

```
##
## 载入程辑包：'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(gutenbergr)
library(tidytext)
library(sentimentr)
```

Task 1

`Gods of the North` is the book i choosed for this assignment I am going to analysis this book by the method mentioned by the Chapter 2 of Text Mining with R do a sentiment display through the narrative of this book.

Task 2

```
north<-gutenberg_download(42664)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

```
tidy_books <- north %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE)))) %>%
  unnest_tokens(word, text)
afinn <- tidy_books %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")
```

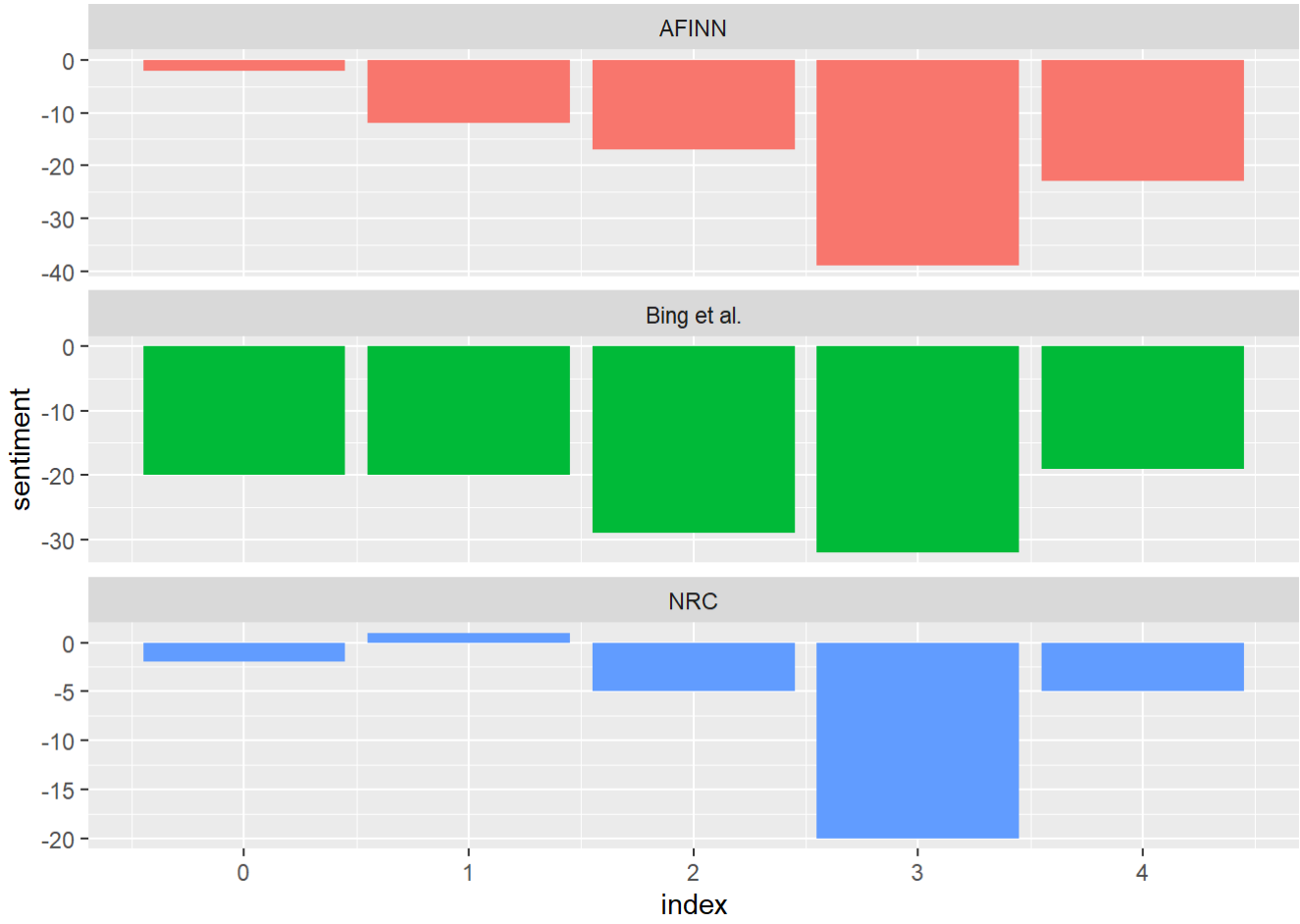
```
## Joining, by = "word"
```

```
bing_and_nrc <- bind_rows(
  tidy_books %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  tidy_books %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
## Joining, by = "word"
```

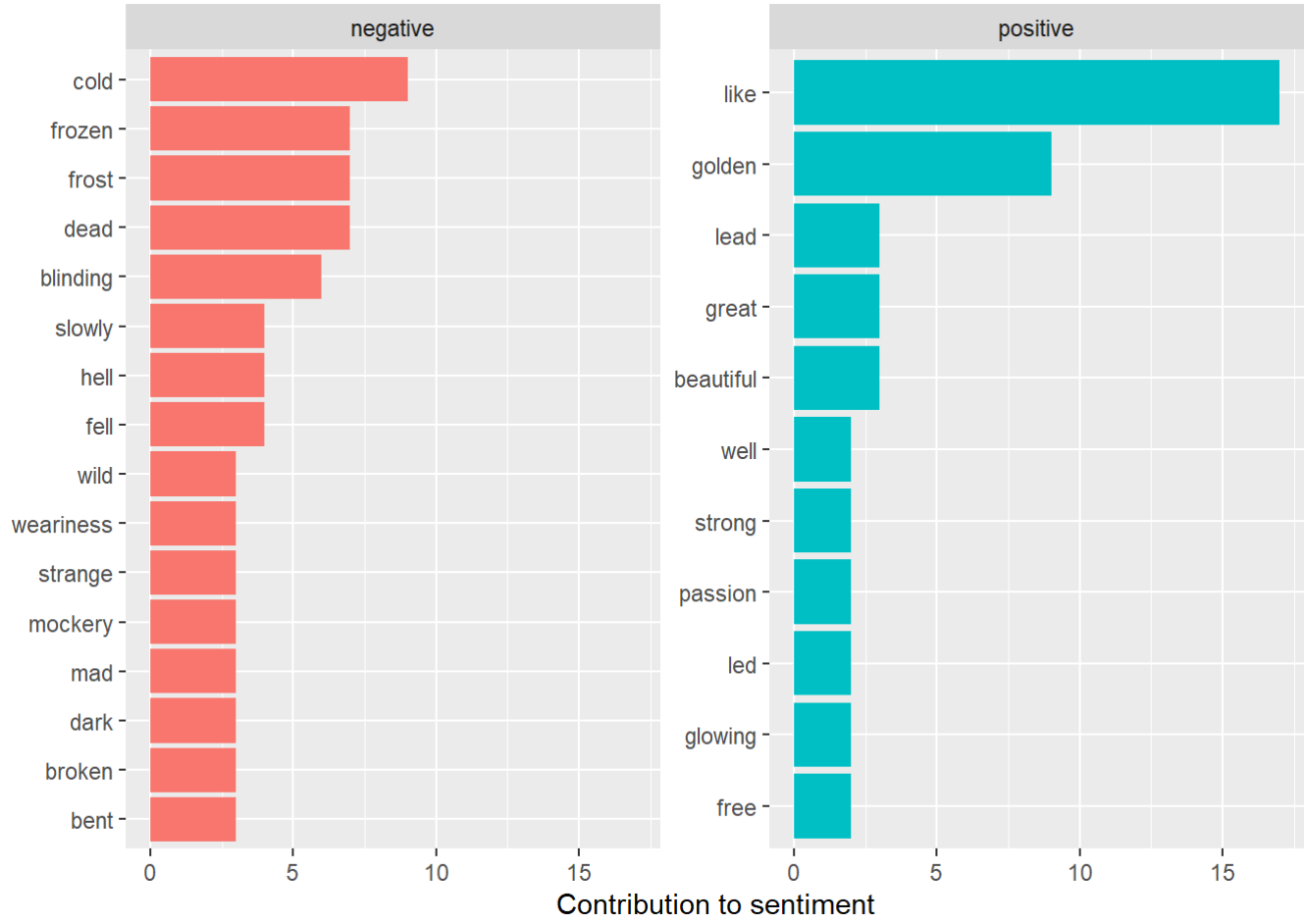
```
bind_rows(afinn,
          bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```




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

```
## Joining, by = "word"
```

```
bing_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
```

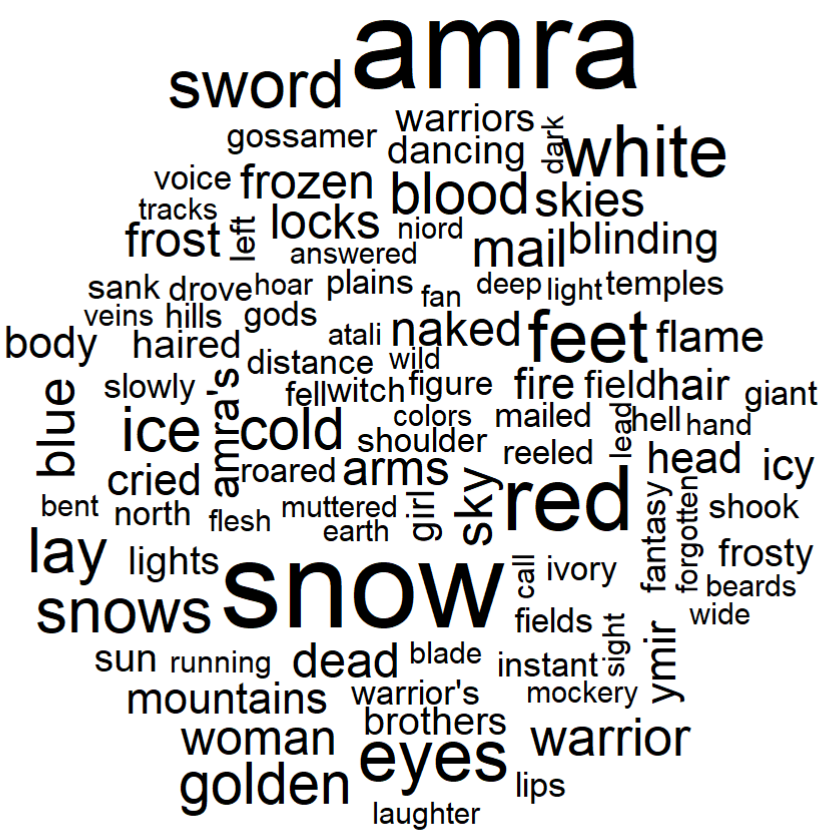



```
library(wordcloud)
```

```
## 载入需要的程辑包：RColorBrewer
```

```
tidy_books %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

```
## Joining, by = "word"
```




```
library(reshape2)
```

```
##
## 载入程辑包：'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                   max.words = 100)
```

```
## Joining, by = "word"
```

negative

positive

```
# bingnegative <- get_sentiments("bing") %>%
#   filter(sentiment == "negative")
#
# wordcounts <- tidy_books %>%
#   group_by(book, chapter) %>%
#   summarize(words = n())
#
# tidy_books %>%
#   semi_join(bingnegative) %>%
#   group_by(book, chapter) %>%
#   summarize(negativewords = n()) %>%
#   left_join(wordcounts, by = c("book", "chapter")) %>%
#   mutate(ratio = negativewords/words) %>%
#   filter(chapter != 0) %>%
#   slice_max(ratio, n = 1) %>%
#   ungroup()
```

Task 3

```
# devtools::install_github("Truenumbers/tnum/tnum")
# library(tnum)
# tnum.authorize("mssp1.bu.edu")
# tnum.setSpace("test2")
# source("Book2TN-v6A-1.R")
# tnBooksFromLines(north$text, "Howard/test2")
# tnum.getDBPathList(taxonomy="subject", levels=2)
#
#  q111 <- tnum.query(query = "north# has ordinal", max=500)   ## everything
#  df111 <- tnum.objectsToDf(q111)
#
#
#
# ## show ordered objects in document
# q112 <- tnum.query("north# has ordinal")   ## show ordered objects in document
# df112 <- tnum.objectsToDf(q112)
#
# ## focus on one paragraph -- note the word count for each sentence
# q3 <- tnum.query("north/test2/chapter-1/paragraph-1# has count#")  # just 1 para
# df3 <- tnum.objectsToDf(q3)
# df3
#
#
# ## and now look at the text in a sentence
# q1 <- tnum.query("north/test2/chapter-1/paragraph-1/sentence-3# has text")
# df1 <- tnum.objectsToDf(q1)
# df1
#
# ## to see the text
# q3 <- tnum.query("north/test2/chapter-1/paragraph-1sentence-3/ has string.value")
# unlist(q3)
# q20 <- tnum.query("north/test2# has *", max=3)
# df20 <- tnum.objectsToDf(q20)
#
#
# q24 <- tnum.query("north/test2/heading# has *", max=60)
# df24 <- tnum.objectsToDf(q24)
#
#
# q22 <- tnum.query("north/test2/heading:0022# has *")
# df22 <- tnum.objectsToDf(q22)
# ord_ch1 <- unlist(tnum.query("north/test2/heading:0022# has ordinal"))
# ord_ch2 <- unlist(tnum.query("north/test2/heading:0023# has ordinal"))
#
#
# q25 <- tnum.query("north/test2/heading:0023# has *")
# df25 <- tnum.objectsToDf(q25)
#
#
# ch1_txt <- tnum.query("north/test2/section:0022/paragraph:0002/# has text", max=30)
# ch1_txt_df <- tnum.objectsToDf(ch1_txt)
# ch1_txt_df$string.value
#
#
#
```

```
# ch2_txt <- tnum.query("north/test2/section:0022/paragraph:0002/sentence:# has *", max=30)
# ch2_txt_df <- tnum.objectsToDf(ch2_txt)
#
# ch2_txt_df$string.value
#
# length(ch2_txt_df$string.value)
#
#
# q21 <- tnum.query("north/test2/section:0022/paragraph:0001/# has *", max = 30)
# df21 <- tnum.objectsToDf(q21)
```