# task3

Handing Zhang

12/10/2021

## Task 3

```
# devtools::install_github("Truenumbers/tnum/tnum")
# library(Truenumbers) this one did not work
library(tnum)
```

```
pacman::p_load(
  gutenbergr,
  tidytext,
  magrittr,
  textdata,
  dplyr,
  stringr,
  tidyverse,
  tidyr,
  scales,
  reshape2,
  ggplot2,
  tinytex,
  latexpdf,
  sentimentr)
```

Create tidy form of ***game*** as we did in task2.

```
game <- gutenberg_download(1160)
```

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

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

```
#
# tidy_game <- game %>%
#   unnest_tokens(word, text) %>%  # output is word column, input is from text column in original game
#   anti_join(stop_words) # get rid of stop words
#
#

tidy_game <- game %>%
```

```r
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("chapter",
                                      ignore_case = TRUE)))) %>%
  unnest_tokens(word, text)
```

```r
## first create my own branch for the text into the mssp server
source('Book2TN-v6A-1.R')
tnum.authorize('mssp1.bu.edu') # get the access of the server
```

```
## Available spaces: testspace, MEPED, alion-rf, shared-testspace, test2, alion, NCNM, ED-900-Workshop,
```

```
## Numberspace set to: testspace
```

```r
tnum.setSpace("test3") # use the test3 space of the server
game_txt <- readLines("game.txt") #game 1160
# tnBooksFromLines(game_txt, 'handing/game2')  # use the Book2Tn to digest (started 10:27)
tnum.getDBPathList(taxonomy = 'subject', level = 2) # check the branch in server
```

```
##  [1] ""                                "lewiscarrol/alice"
##  [3] "carrol/alice"                    "carroll/alice"
##  [5] "elisa/the_call_of_the_wild"      "handing/hw4"
##  [7] "zara/hw4"                        "handing/sea"
##  [9] "zara/A4"                         "zara/a4"
## [11] "elisa/wild"                      "dostoevsky/hw4"
## [13] "dostoevsky/crime_and_punishment" "handing/game1"
## [15] "handing/game2"                   "sisitzky/scarlet"
## [17] "william/test3"                   "zara/homework4"
## [19] "zara/submission4"
```

```r
DF6<- tnum.query('handing/game2/section# has text',max=10000) %>% tnum.objectsToDf()
```

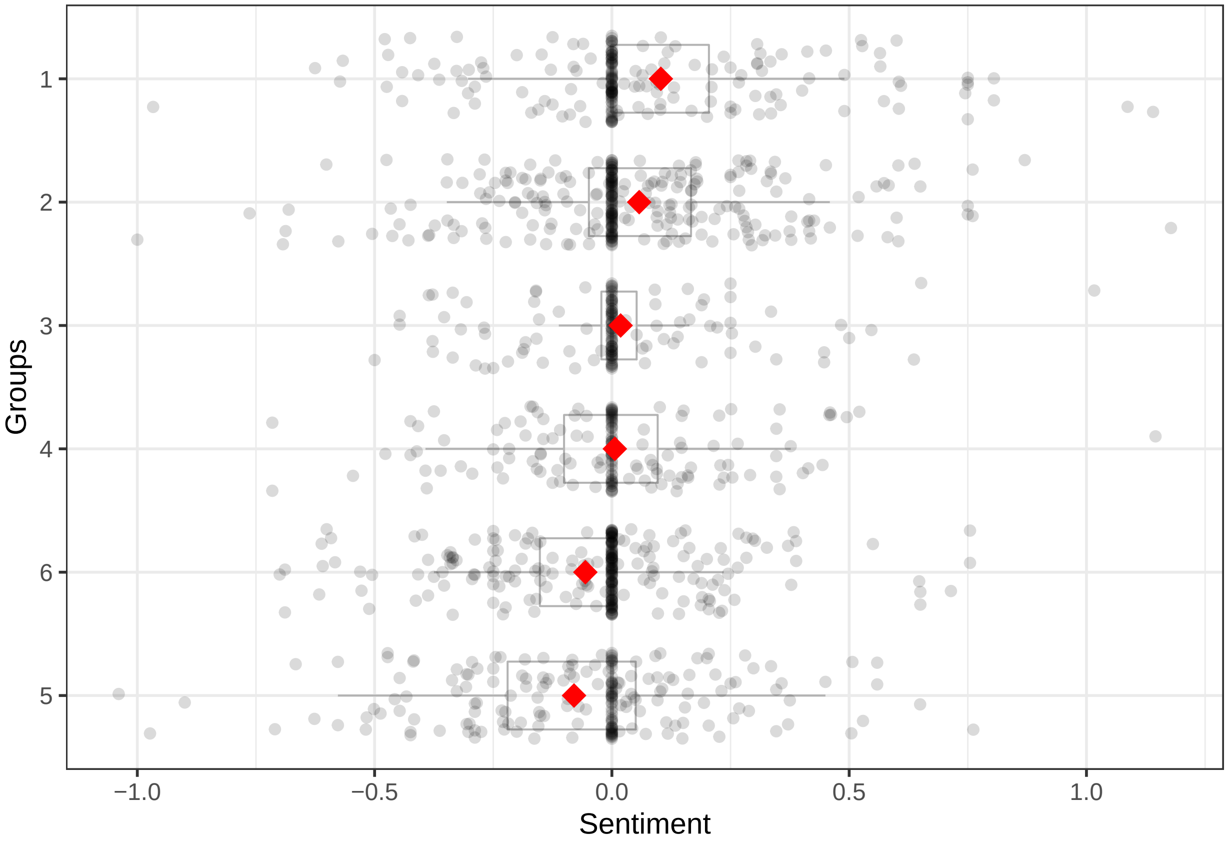
```
## Returned 1 thru 867 of 867 results
```

```r
game_sentence<-DF6 %>% separate(col=subject,
                  into = c("path1", "path2","section","paragraph","sentence"),
                  sep = "/",
                  fill = "right") %>%
  select(section:string.value)
```

```r
#book_sentence$section<-str_extract_all(book_sentence$section,"\\d+") %>% unlist() %>% as.numeric()
game_sentence <- game_sentence %>% mutate_at(c('section','paragraph','sentence'),~str_extract_all(.,"\\
```

```r
sentence_out <- game_sentence %>% dplyr::mutate(sentence_split = get_sentences(string.value)) %$%
    sentiment_by(sentence_split, list(section))

plot(sentence_out)
```



This picture shows the sentiments number in each section and lists the number of sentiments words in each section. The range of x-axis is from -1 to 1. Dots in -1 to 1 mean negative words, in the contrary, range 0 to 1 contains positive words. Based on the density of dots, the result is clear that there are more positive words than negative words in this book, which corresponds to the word cloud in task two. In addition, sention 7 contains the most number of sentiment words and section 1 has the less.

### Compare two methods that were ultilized in Task Two and Task Three.

```
# create a new bing with index=chapter
new_bing<-tidy_game %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al.") %>%
    count(method, index = chapter, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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

```
# scale sentiment to keep unit same
new_bing2<-new_bing %>%
  mutate(bing_scale=scale(sentiment)) %>%
  select(method,index,bing_scale)
```

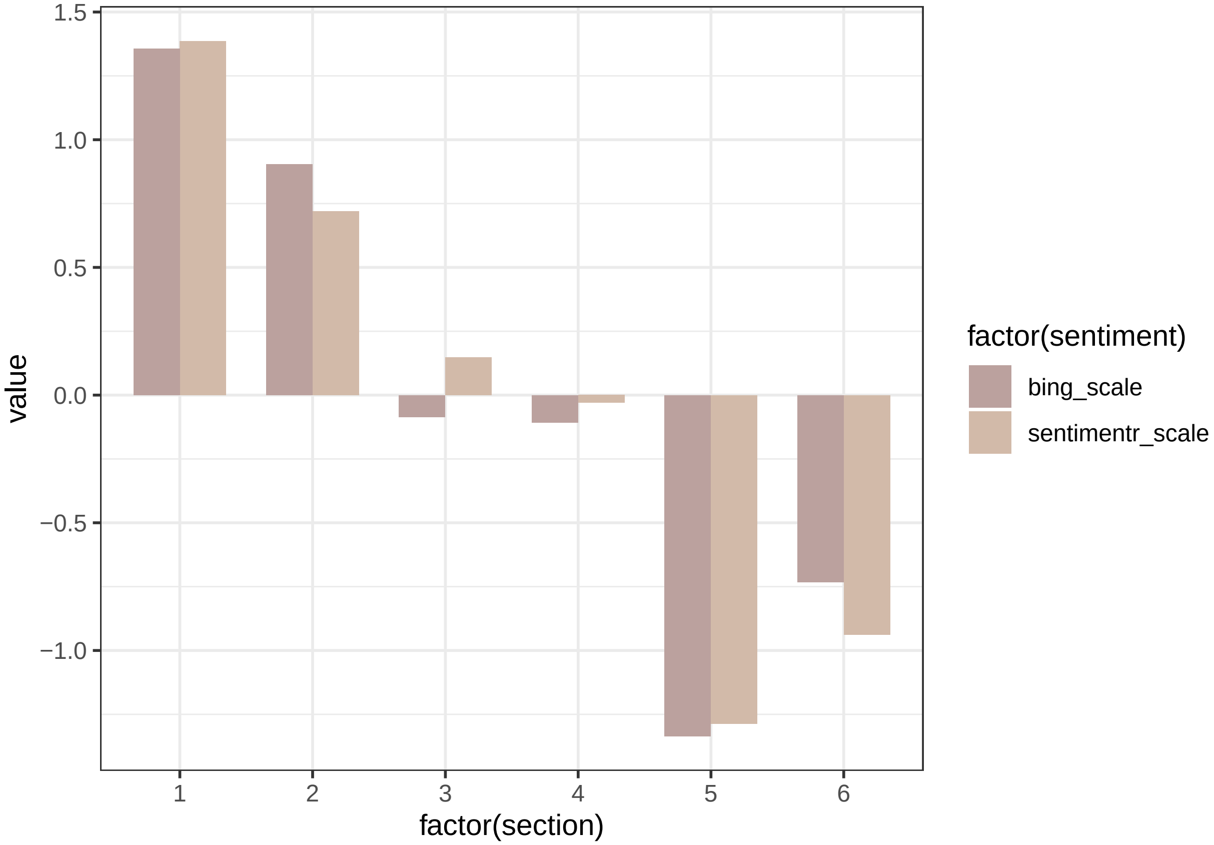
```
# change colname in order to join by section
colnames(new_bing2)[2]='section'

# scale sentiment to keep unit same
sentence_out<-sentence_out %>% mutate(sentimentr_scale=scale(ave_sentiment))

# join two df together
sentence_out_2method<-left_join(sentence_out,new_bing2,by='section')%>% select(section,bing_scale,senti

# use pivot longer for ggplot
sentence_out_2method_plot<-sentence_out_2method %>% pivot_longer(cols=c('sentimentr_scale','bing_scale'

# create a barplot to compare two methods
sentence_out_2method_plot %>%ggplot(aes(y=value,x=factor(section))) +
  geom_bar(aes(fill=factor(sentiment)),stat='identity',position = "dodge",width = 0.7)+theme_bw()+
  scale_fill_manual('factor(sentiment)',values=c("#bba19e","#d2baa9"))
```



Due to these are two different methods, it is not easy and reasonable to compare them directly. Therefore, I limited the range of these sentiment words, just similar to what I did in the previous diagram in task 2. After defining the scale, I made a bar plot to explain the result. In each session, the sentiment trends in vocabulary are roughly the same, but the specific values are different. However, I think setimentr method is better the Bing.

## Reference:

1.https://github.com/MA615-Yuli/MA615_assignment4_new 2.https://www.gutenberg.org/ebooks/1160 3.https://www.tidytextmining.com/sentiment.html