Solution slides Part 1 Introduction to R & Data for Humanities

Afternoon session

Text-mining with Tidyverse

Exercise 7

7a.

```
```{r}
# Exercise 7a.Now let's start tidy-texting Dickinson's poem. Run the following lines of code:

text <- c("Because I could not stop for Death -",
          "He kindly stopped for me -",
          "The Carriage held but just Ourselves -",
          "and Immortality")
text
```
```

```
[1] "Because I could not stop for Death -"   "He kindly stopped for me -"           "The Carriage held but just Ourselves -"
[4] "and Immortality"
```

7b.

```{r}
# Excercise 7b. Let's call on a package from the Tidyverse that will give us the right data frame: dplyr. You can call on this package by running the following code:


library(dplyr)
text_df <- tibble(line = 1:4, text = text)

text_df
```

line <int>	text <chr>
1	Because I could not stop for Death -
2	He kindly stopped for me -
3	The Carriage held but just Ourselves -
4	and Immortality

4 rows

7c.

```{r}
# Excercise 7c. We will now break the text into individual tokens (tokenization) and transform it to a tidy data structure. To do this, call on the unnest_tokens() function:

library(tidytext)

text_df %>%
  unnest_tokens(word, text)
```

line <int>	word <chr>
1	because
1	i
1	could
1	not
1	stop
1	for
1	death
2	he
2	kindly
2	stopped

1-10 of 20 rows

Previous 1 2 Next

8a.

```r
# Excercise 8a. Based on the previous exercises with Dickenson's poem, are you now able to call on the janeaustenr package, as well as the dplyr and stringr packages needed for your analysis?

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

```
package 'janeaustenr' was built under R version 4.0.5package 'stringr' was built under R version 4.0.5
```

← Don't worry about the warnings, the packages should work just fine!

8b.

```r
# Excercise 8b. Run the following code… and then challenge yourself with exercise 8c!

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

original_books
```

text <chr>	book <fctr>	linenumber <int>
	Sense & Sensibility	31
By a former marriage, Mr. Henry Dashwood had one son: by his present	Sense & Sensibility	32
lady, three daughters. The son, a steady respectable young man, was	Sense & Sensibility	33
amply provided for by the fortune of his mother, which had been large,	Sense & Sensibility	34
and half of which devolved on him on his coming of age. By his own	Sense & Sensibility	35
marriage, likewise, which happened soon afterwards, he added to his	Sense & Sensibility	36
wealth. To him therefore the succession to the Norland estate was not	Sense & Sensibility	37
so really important as to his sisters; for their fortune, independent	Sense & Sensibility	38
of what might arise to them from their father's inheriting that	Sense & Sensibility	39
property, could be but small. Their mother had nothing, and their	Sense & Sensibility	40

31-40 of 73,422 rows | 1-3 of 4 columns

Previous 1 2 3 4 5 6 … 100 Next

When you see the output at first, you see a lot of blank lines in the ‘text’ column. This is because the title page is also taken into account. When you browse through the table, you will soon encounter actual lines of text.

If you press the arrow next to the ‘linenumber’ column, you see the number of the chapter the line is in.

8c.

```r
# Excercise 8c

library(tidytext)
tidy_books <- original_books %>%
  unnest_tokens(word, text)

tidy_books
```

book <fctr>	linenumber <int>	chapter <int>	word <chr>
Sense & Sensibility	1	0	sense
Sense & Sensibility	1	0	and
Sense & Sensibility	1	0	sensibility
Sense & Sensibility	3	0	by
Sense & Sensibility	3	0	jane
Sense & Sensibility	3	0	austen
Sense & Sensibility	5	0	1811
Sense & Sensibility	10	1	chapter
Sense & Sensibility	10	1	1
Sense & Sensibility	13	1	the

1-10 of 725,055 rows

Previous 1 2 3 4 5 6 ... 100 Next

8d.

```r
# Excercise 8d. This is the first time you call on a dataset from a package. Can you call on the anti_join() function by completing the code below?


data(stop_words)

tidy_books <- tidy_books %>%
  anti_join(stop_words)

# We can also use dplyr's count() to find the most common words in all the books as a whole.
tidy_books %>%
  count(word, sort = TRUE)

```

word <chr>	n <int>
miss	1855
time	1337
fanny	862
dear	822
lady	817
sir	806
day	797
emma	787
sister	727
house	699

1-10 of 13,914 rows

8e.

```{r}
# Excercise 8e. Let's build ourselves a pipeline! Run this code and see what happens…

library(ggplot2)

tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 600) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL)
```