What

This is a package that does/has only one thing: the complete transcriptions of all episodes of The Office! (US version).

Use this data set to master NLP or text analysis. Let’s scratch the surface of the subject with a few examples

First install the package from CRAN:

# install.packages("schrute") library(schrute)

There is only one data set with the schrute package; assign it to a variable

mydata <- schrute::theoffice

Take a peek at the format:

dplyr::glimpse(mydata) #> Observations: 55,130

#> Variables: 7

#> $ index 1, 358, 715, 1072, 1429, 1786, 2143, 2500, 2857…

#> $ season "01", "01", "01", "01", "01", "01", "01", "01",…

#> $ episode "01", "01", "01", "01", "01", "01", "01", "01",…

#> $ episode\_name " Pilot", " Pilot", " Pilot", " Pilot", " Pilot… #> $ character "Michael", "Jim", "Michael", "Jim", "Michael", … #> $ text " All right Jim. Your quarterlies look very goo… #> $ text\_w\_direction " All right Jim. Your quarterlies look very goo…

mydata %>%

dplyr::filter(season == '01') %>% dplyr::filter(episode == '01') %>% dplyr::slice(1:3) %>% knitr::kable()

# index season episode episode\_name character text text\_w\_direction

All right Jim. Your

1 01 01 Pilot Michael

358 01 01 Pilot Jim

715 01 01 Pilot Michael

quarterlies look very good. How are things at the library?

Oh, I told you. I couldn’t close it. So…

So you’ve come to the master for guidance? Is this what you’re saying, grasshopper?

All right Jim. Your quarterlies look very good. How are things at the library?

Oh, I told you. I couldn’t close it. So…

So you’ve come to the master for guidance? Is this what you’re saying, grasshopper?

So what we have is the season, episode number and name, character, the line spoken and the line spoken with the stage direction (cue).

We can tokenize all of the lines with a few lines from the tidytext package:

token.mydata <- mydata %>% tidytext::unnest\_tokens(word, text)

This increases our data set to 575146 records, where each record contains a word from the script.

token.mydata %>%

dplyr::filter(season == '01') %>% dplyr::filter(episode == '01') %>% dplyr::slice(1:3) %>% knitr::kable()

# index season episode episode\_name character text\_w\_direction word

All right Jim. Your quarterlies look very good. How are

|  |  |  |  |
| --- | --- | --- | --- |
| 1 01 | 01 | Pilot | Michael |
| 1 01 | 01 | Pilot | Michael |
| 1 01 | 01 | Pilot | Michael |

things at the library?

All right Jim. Your quarterlies look very good. How are things at the library?

All right Jim. Your quarterlies look very good. How are things at the library?

all right jim

If we want to analyze the entire data set, we need to remove some stop words first:

stop\_words <- tidytext::stop\_words

tidy.token.mydata <- token.mydata %>% dplyr::anti\_join(stop\_words, by = "word")

And then see what the most common words are:

tidy.token.mydata %>% dplyr::count(word, sort = TRUE)

#> # A tibble: 19,225 x 2 #> word n

#>

#> 1 yeah 2895

#> 2 hey 2189

#> 3 michael 2054

#> 4 dwight 1540

#> 5 uh 1433

#> 6 gonna 1365

#> 7 jim 1345

#> 8 pam 1168

#> 9 time 1129

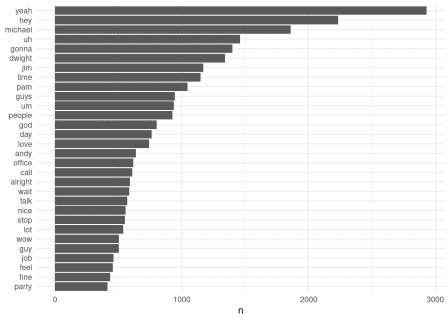
#> 10 guys 933

#> # … with 19,215 more rows

tidy.token.mydata %>% dplyr::count(word, sort = TRUE) %>% dplyr::filter(n > 400) %>%

dplyr::mutate(word = stats::reorder(word, n)) %>% ggplot2::ggplot(ggplot2::aes(word, n)) + ggplot2::geom\_col() +

ggplot2::xlab(NULL) + ggplot2::coord\_flip() + ggplot2::theme\_minimal()



Feel free to keep going with this. Now that you have the time line (episode, season) and the character for each line and word in the series, you can perform an unlimited number of analyses. Some ideas:

* Sentiment by character
* Sentiment by character by season
* Narcissism by season (ahem.. Nard Dog season 8-9)
* Lines by character
* Etc.