Text Mining and Sentiment Analysis

Prof. Annamaria Bianchi
A.Y. 2024/2025

Lecture 6

4 March 2025

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Scienze Economiche

Outline

Initial case study

Package: **stringr**

Projects

Useful when:

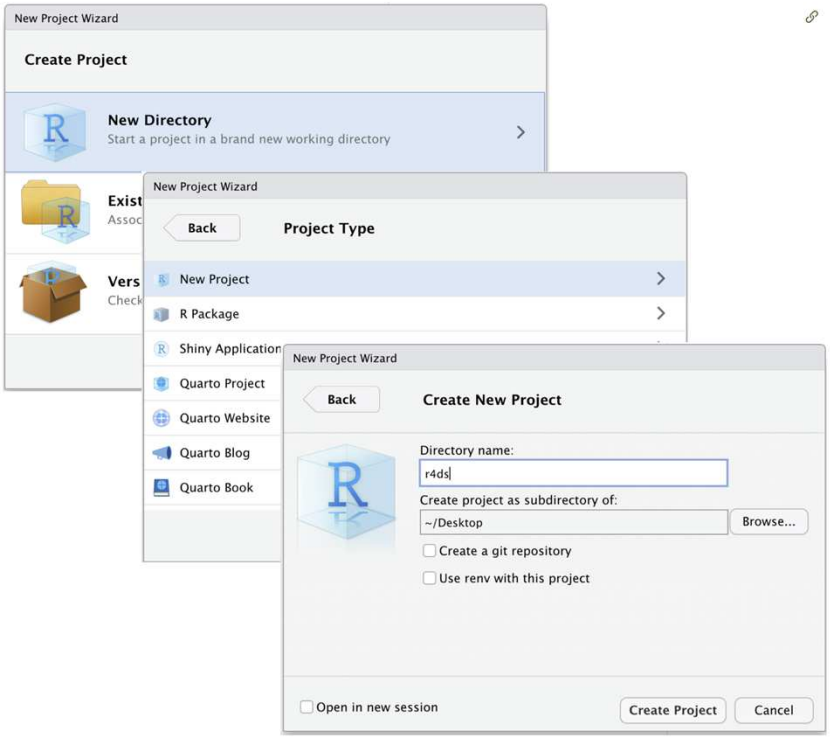
- you quit R, go do something else, and return to your analysis later.
- you are working on multiple analyses simultaneously and you want to keep them separate.
- you need to bring data from the outside world into R and send numerical results and figures from R back out into the world

We have already learnt to set the working directory.
There's a better way: the RStudio projects.

Keeping all the files associated with a given project (input data, R scripts, analytical results, and figures) together in one directory is such a wise and common practice that RStudio has built-in support for this via **projects**.
Let's make a project for you to use while you're working through this course.

Projects

Click File > New Project, then follow the steps



Once this process is complete, you'll get a new RStudio project just for this course. Check that the "home" of your project is the current working directory, entering the command on a script editor and save the file, calling it "Lecture6.R".

Projects

Quit RStudio. Inspect the folder associated with your project — notice the .Rproj file.

Double-click that file to re-open the project.

Notice you get back to where you left off: it's the same working directory and command history, and all the files you were working on are still open. In this way, you will have a completely fresh environment, guaranteeing that you're starting with a clean slate.

Initial case study

Let us assume that we need to launch an **airline competitive customer** service team but we know nothing about the domain or how expensive this customer service channel is.
Referring to the customer service provided on *Twitter* (now *X*) by *Delta Airlines*, we want to obtain some useful information.

Questions:

1) What is the average length of a social customer service reply?
2) How many people should be on a social media customer service team? How many social replies are reasonable for a customer service representative to handle?
3) How help is provided (link with helpful information, direct messages, phone numbers)? In case of links, what links were referenced most often?

Dataset

Delta tweets from Twitter API from October 1 to October 15, 2015 were retrieved. It has been cleaned up and organized from a JSON object with many parameters to a smaller CSV with only tweets and date information.

Data are contained in the file 'oct_delta.csv'

Variables:

weekday	day of the week
month	month
data	day of the month
year	year
text	tweet text

weekday	month	date	year	text
Thu	Oct	1	2015	@mjdout I know that can be frustrating..we hope to have you parked and deplaned shortly. Thanks for your patience. *AA
Thu	Oct	1	2015	@rmarkerm Terribly sorry for the inconvenience. If we can be of assistance at this time, pls let us know. *AA
Thu	Oct	1	2015	@checho85 I can check, pls follow and DM your confirmation # for review. *AA
Thu	Oct	1	2015	@nealaa ...Alerts, pls check here: http://t.co/0jlcZnT95Q *JH 3/3

Introducing the data in R

Let us read the data into R

```
> library(tidyverse)
> getwd()
> text.tbl = read_csv("oct_delta.csv")
Rows: 1377 Columns: 5
── Column specification ────────────────────────────────────────
Delimiter: ","
chr (3): weekday, month, text
dbl (2): date, year
```

Data characteristics

And have a look at the main characteristics of the data frame

```
> class(text.tbl)
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
> glimpse(text.tbl)
Rows: 1,377
Columns: 5
$ weekday <chr> "Thu", "Thu", "Thu", "Thu", "Thu", "Thu", "Thu", "Thu",~
$ month   <chr> "Oct", "Oct", "Oct", "Oct", "Oct", "Oct", "Oct", "Oct",~
$ date    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ year    <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2~
$ text    <chr> "@mjdout I know that can be frustrating..we hope to hav~
> View(text.tbl)
```

First question

Q1: What is the average length of a social customer service reply?

To answer the first question we might use the function `str_length()` which returns the number of characters in a string. Recall that the function is vectored (meaning that it can be applied to a character column directly) and that it does count spaces as characters.

```
>str_length(text.tbl[4,5])
[1] 65
> str_length(text.tbl$text)
[1] 119 110 78 65 137 142 75 44 140 141 54 62
[13] 59 103 136 111 128 73 69 100 74 116 117 61
….
> mean(str_length(text.tbl$text))
[1] 92.14452
```

Answer: The average length of a social customer service reply is approximately 92 characters. Since tweets can be a maximum of 140 characters, the insight here is that agents are concise and not often maximizing the Twitter character limit.
Notice that in the dataset there are cases of a long message being broken up into multiple tweets.

Splitting

`str_split()` Split up a string into pieces.

```
str_split(string, pattern, simplify = F)
str_replace_all(string, pattern, simplify = F)
```

`string`	A character vector.
`pattern`	Pattern to look for (regular expression).
`simplify`	If `FALSE`, the default, returns a list of character vectors. If `TRUE` returns a character matrix.

```
> fruits <- c(
+     "apples and oranges and pears and bananas",
+     "pineapples and mangos and guavas"
+ )
>
> str_split(fruits, " and ")
[[1]]
[1] "apples"  "oranges" "pears"   "bananas"

[[2]]
[1] "pineapples" "mangos"     "guavas"
```

Second question

Q2: How many people should be on a social media customer service team? How many social replies are reasonable for a customer service representative to handle?

Let us first have a look at the first two tweets of the dataframe

```
> text.tbl$text[1:2]
[1] "@mjdout I know that can be frustrating..we hope to have you parked and
 deplaned shortly. Thanks for your patience.  *AA"
[2] "@rmarkerm Terribly sorry for the inconvenience. If we can be of assista
nce at this time, pls let us know.  *AA"
```

We can see that agents are adding personal initials to each tweet. In the first two tweets initials are separated by an * from the rest.

Second question

First idea: use the function `str_split()` to identify the agent for each tweet. Recall that the function `str_split()` creates subset strings by matching character patterns.

```
> str_split(text.tbl$text[1:2], "\\*")
[[1]]
[1] "@mjdout I know that can be frustrating..we hope to have you parked and
 deplaned shortly. Thanks for your patience.  "
[2] "AA"


[[2]]
[1] "@rmarkerm Terribly sorry for the inconvenience. If we can be of assista
nce at this time, pls let us know.  "
[2] "AA"
```

The result is a list with each second value holding the information we are interested in. Here the same agent 'AA' signed both tweets.

Second question

Using the `simplify` argument, we could create the matrix version

```
> str_split(text.tbl$text, "\\*", simplify=T)[1:5,]
     [,1]

[1,] "@mjdout I know that can be frustrating..we hope to have you parked and
 deplaned shortly. Thanks for your patience.  "
[2,] "@rmarkerm Terribly sorry for the inconvenience. If we can be of assist
ance at this time, pls let us know.  "
[3,] "@checho85  I can check, pls follow and DM your confirmation # for revi
ew.  "
[4,] "@nealaa ...Alerts, pls check here: http://t.co/0jlcZnT95Q "

[5,] "@nealaa ...advisory has only been issued for the Bahamas, but that cou
ld change.  To check for updates on Weather advisories &amp;amp;... 2/3"
     [,2]     [,3]
[1,] "AA"     ""
[2,] "AA"     ""
[3,] "AA"     ""
[4,] "JH 3/3" ""
[5,] ""       ""
```

Second question

The above code works as long as all agents are using the same pattern to close their messages. If an agent uses another character such as a dash instead of an asterisk, then the `str_split()` function would miss that tweet signature. It may be the case that agents use a mixture of patterns to close messages.

How could we identify the signatures of the agents then??

Second question

We may try to capture the final two characters from each tweet.

The function `str_sub()` extracts substrings from a character vector depending on their position. It takes start and end arguments that give the (inclusive) position of the substring. In the present case, we need to extract the last two characters.

With reference to the first two tweets:

```
> str_sub(text.tbl$text[1:2], -2, -1)
[1] "AA" "AA"
```

Second question

Let us now create a new variable in the dataframe containing the agent signatures

```
> text.tbl <- text.tbl |>
+ mutate( + agents = str_sub(text, -2, -1)
+ )
> View(text.tbl)
```

	weekday	month	date	year	text	agents
1	Thu	Oct	1	2015	@mjdout I know that can be frustrating..we hope to have yo...	AA
2	Thu	Oct	1	2015	@rmarkerm Terribly sorry for the inconvenience. If we can b...	AA
3	Thu	Oct	1	2015	@checho85 I can check, pls follow and DM your confirmati...	AA
4	Thu	Oct	1	2015	@nealaa ...Alerts, pls check here: http://t.co/0jlcZnT95Q *JH ...	/3
5	Thu	Oct	1	2015	@nealaa ...advisory has only been issued for the Bahamas, b...	/3
6	Thu	Oct	1	2015	@nealaa Hi. Our meteorologist team is aware of Hurricane J...	/3
7	Thu	Oct	1	2015	@BigGucciQueen This is your direct dial number + 43 (0)1 ...	DD

Notice that some tweets are continuation of customer service cases and are showing up as '/3' or something similar. Because these are continuations, and we are concerned about individual caseloads, to answer our question we can ignore them as a first step approximation

Second question

Let us compute the frequency table of the agents' signatures and create the corresponding tibble

```
> Freq.agents <- text.tbl |>
+ count(agents)
> Freq.agents
# A tibble: 36 × 2
agents n
<chr> <int>
1 /2   267
2 /3   69
3 /4   11
4 AA   62
5 AB   42
6 AD   12
```

We shall quickly determine that the busiest agent (over the entire period) is PL. We shall also examine the average among all agents.

Second question

Let us drop the first three lines of the table:

```
> Freq.agents = Freq.agents[-c(1,2,3),]
> Freq.agents
```

Next we shall compute the number of people in the social media customer service team:

```
> dim(Freq.agents)
[1] 33 2
```

Second question

Now let us compute the maximum number of replies and the average frequency among all agents

```
> Freq.agents <- Freq.agents |>
+ arrange(desc(n))
> Freq.agents |>
+ summarise(
+ mean(n), max(n)
+ )
# A tibble: 1 × 2
`mean(n)` `max(n)`
<dbl>     <int>
1 31.2         95
```

Answer: The number of agents in the social media team were 33. The hardest working Delta customer service agent was PL with 95 replies. On average, each agent handled 31.2 replies in October. In the week between 5 and 9 October each agent handled 11.69 replies on average

Second question

Exercise. Restrict the analysis to one week, considering only replies between 5th and 9th October.

How many agents were working on that week and how many replies they handled on average?

Third question

Q3: How help is provided (link with helpful information or direct messages)? In case of links, what links were referenced most often?

We need to search for any instance of 'http', in order to identify tweets that contain a url link.

```
> mean(str_detect(text.tbl$text, "http"))
[1] 0.04284677
```

In order to check what links were referenced. This might help to identify whether the links were to an input form or general information.

```
> str_subset(text.tbl$text, "http")
 [1] "@nealaa ...Alerts, pls check here: http://t.co/0jlcZnT95Q *JH 3/3"

 [2] "@owroc ...http://t.co/sNIn5Equux *AA 2/2"
```

Third question

Let us check the use of direct messages

```
> mean(str_detect(text.tbl$text, "DM"))
[1] 0.1314452
```

It looks like DeltaAssist uses links sparingly in favor of direct messages through the Twitter platform.

Are links and DM used at the same time? How shall we check that?

```
> text.tbl$text[str_detect(text.tbl$text, "http") & str_detect(text.tbl$tex
t, "DM")]
character(0)
```

It looks like DeltaAssist never asks for a DM and provides a url link at the same time.

Third question

Let us try to identify tweets containg phone numbers. Phone numbers in the US have the following form xxx-xxx-xxxx

```
> mean(str_detect(text.tbl$text, "[0-9]{3}-[0-9]{3}-[0-9]{4}"))
[1] 0.03703704
```

Clearly one could answer the question by adding three logical variables to the tibble and summarizing them

```
> text.tbl |>
+ mutate(
+ url = str_detect(text, "http"),
+ DM = str_detect(text, "DM"),
+ phone.number = str_detect(text, "[0-9]{3}-[0-9]{3}-[0-9]{4}")) |>
+ summarise(mean(url), mean(DM), mean(phone.number))
# A tibble: 1 × 3
`mean(url)` `mean(DM)` `mean(phone.number)`
<dbl> <dbl> <dbl>
0.0428 0.131 0.0370
```

Answer: DM are more likely to be used than links and phone numbers to be shared.

```
> write_rds(text.tbl, "text.tbl.rds")
```

Save the modified tibble into external file

Exercise for you

Exercise. With reference to the Delta customer service dataset, we want to study the way replies are given.

1) What is the percentage of times Delta customer service agents use the word ‘thank you’? And the word ‘thanks’? Be careful and pay attention to the case of the letters.
2) What is the percentage of times Delta customer service agents use the word ‘thank you’ or the word ‘thanks’?
3) Substitute the word ‘Thanks’ with the word ‘thank you’ in all the tweets in the dataframe.
4) What is the percentage of times Delta customer service agents use the word ‘please’? Notice that sometimes theabbreviation ‘pls’ is used.
5) Substitute the word ‘pls’ with the word ‘please’ in all the tweets in the dataframe.
6) What is the percentage of times Delta customer service agents state that they are sorry or they apologize? Take a look at these sentences.