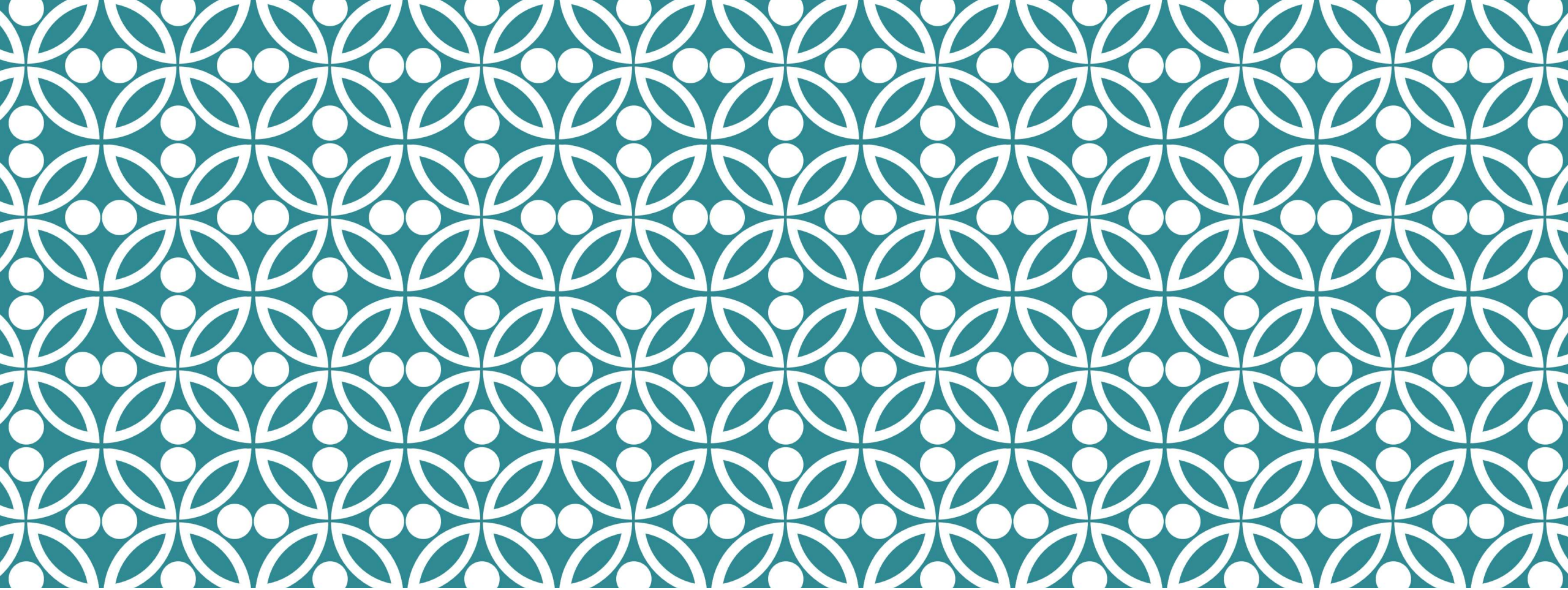


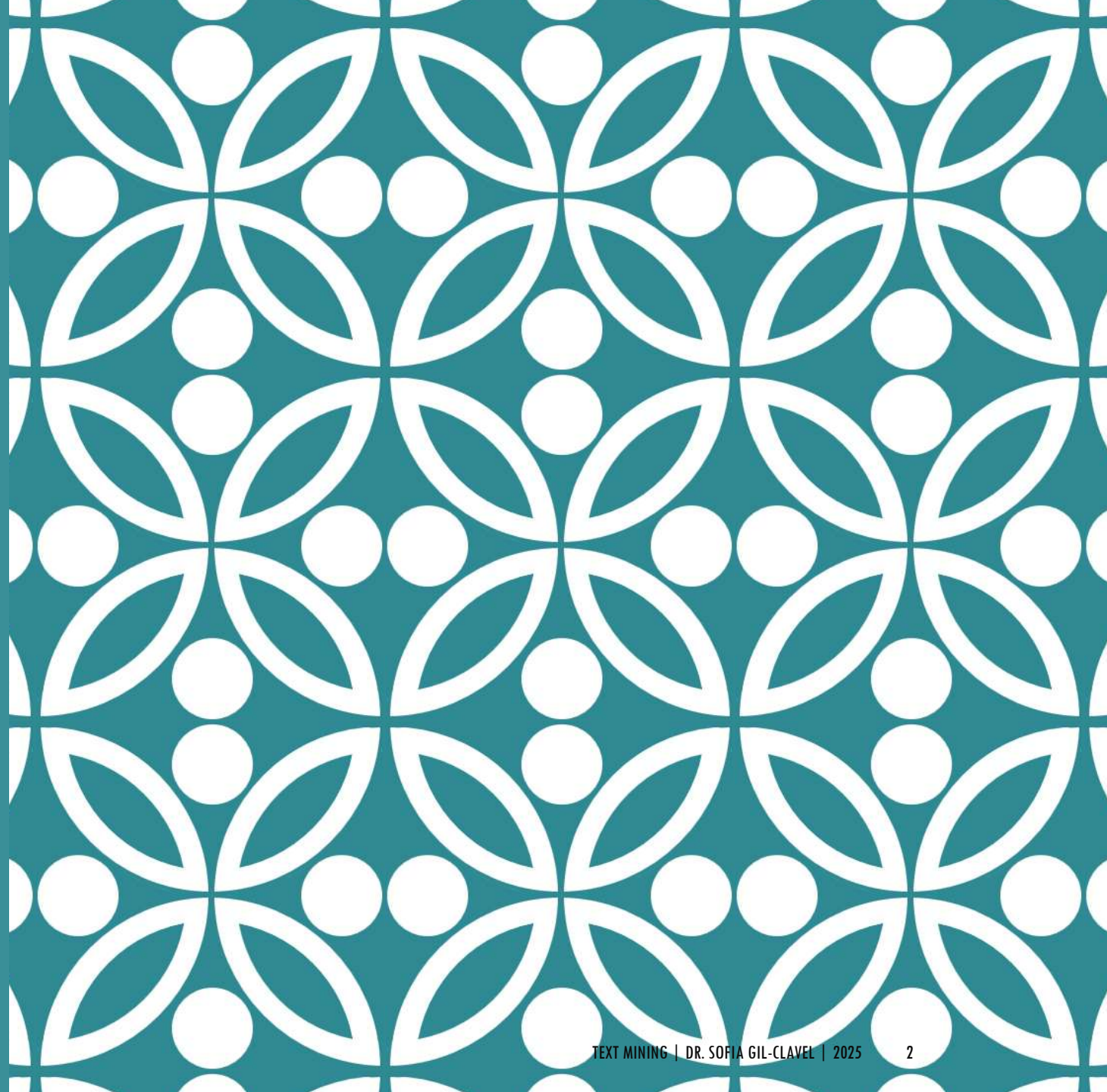
SESSION 8: TEXT MINING

DR. SOFIA GIL-CLAVEL

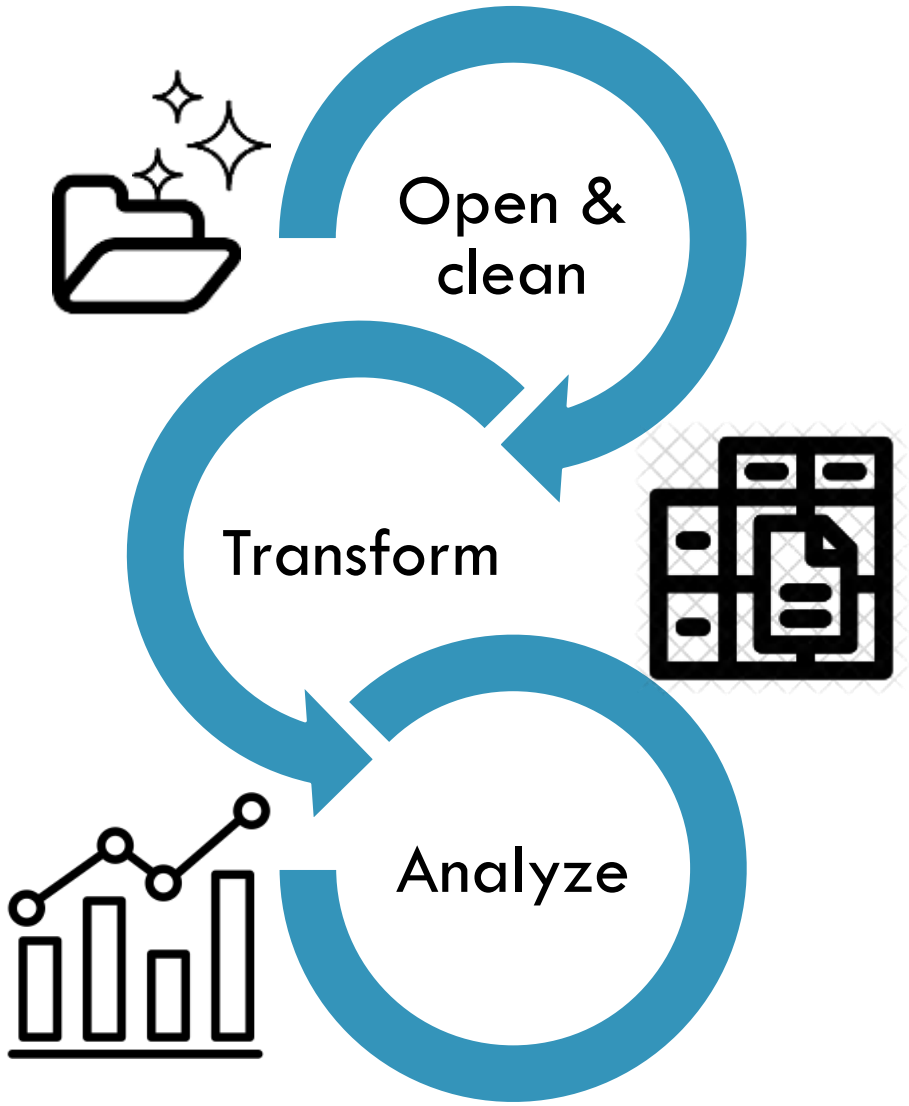
- Recap of the basics of handling text in R.
- Text Mining and Data Viz.
- Quick overview of advanced topics.

1. RECAP BASICS OF HANDLING TEXT IN R

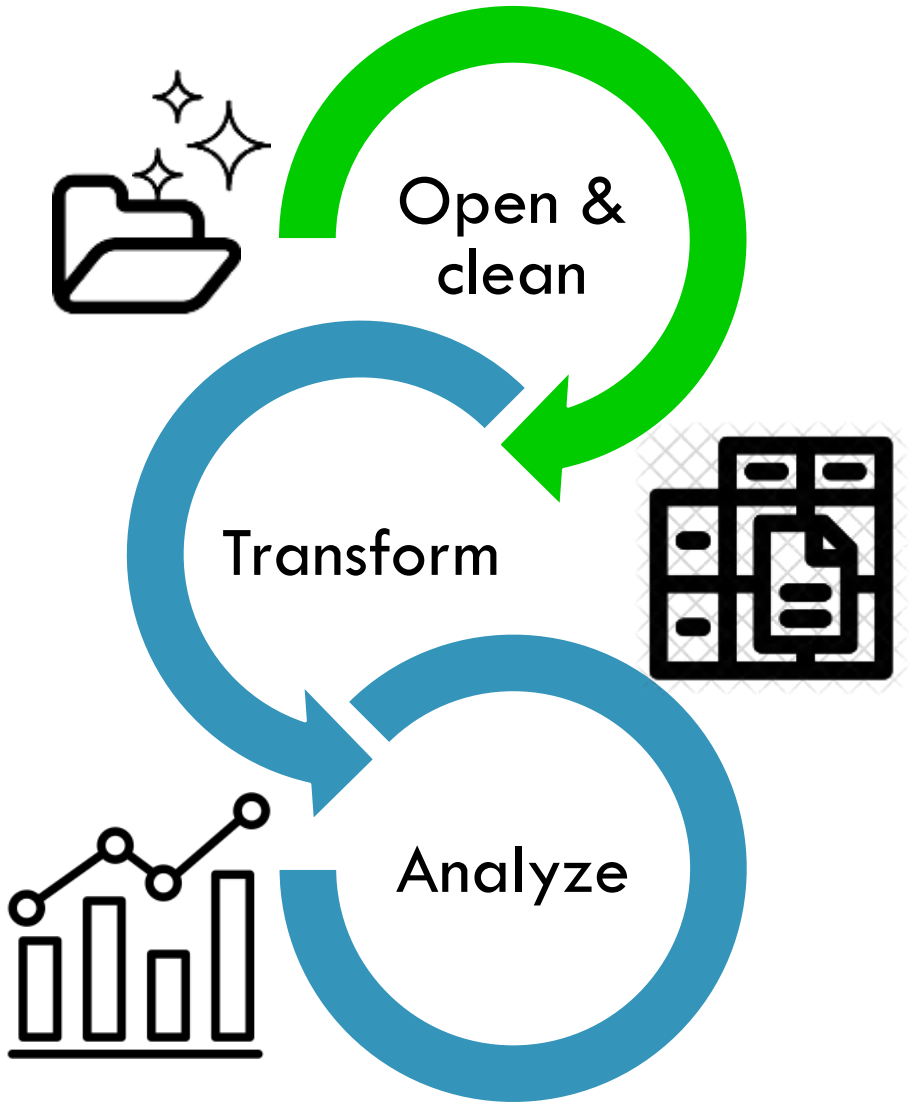
1.1 Basic text handling
1.2 Basic functions
1.3 Tidy text



The pipeline



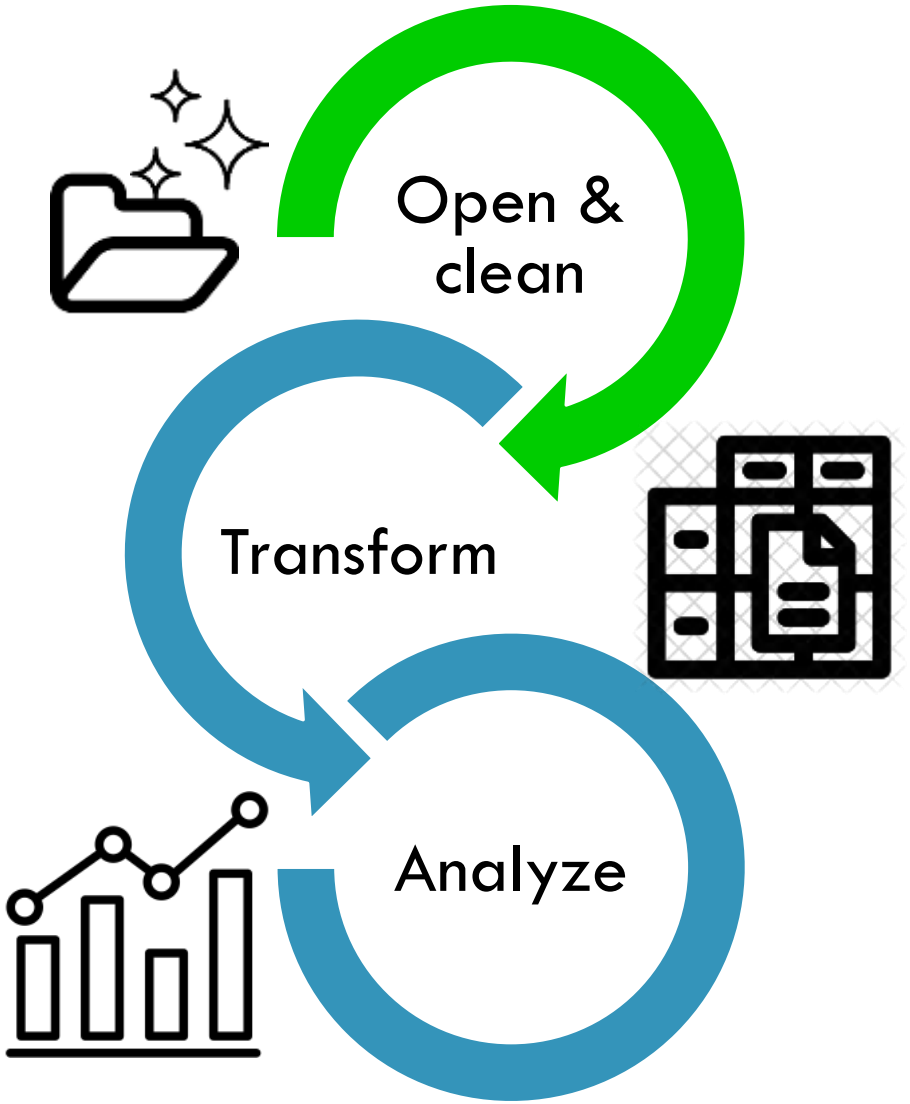
The pipeline



Mention three things that you learned.

The pipeline

```
TXT=read.delim(...)
TXT=tibble(TXT)
```

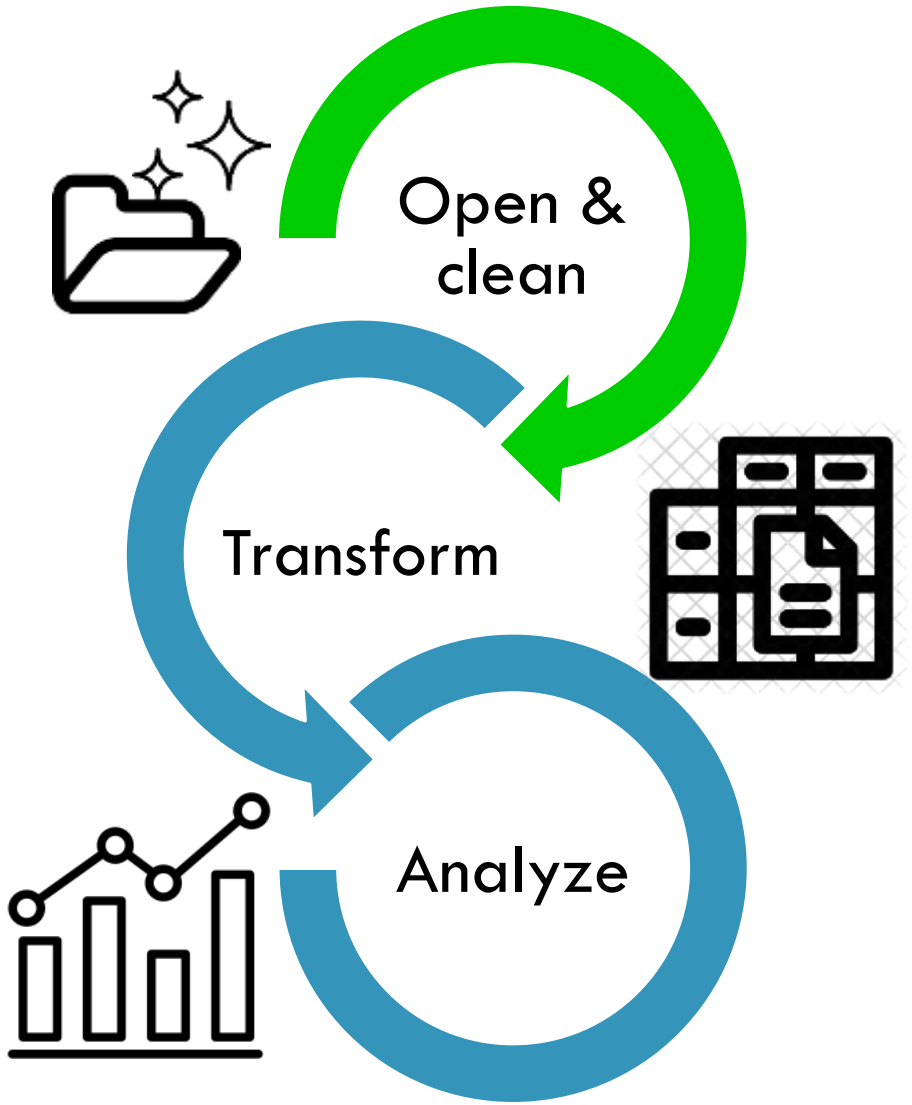


Mention three things that you learned.

The pipeline

Right encoding

```
TXT=read.delim(...)
TXT=tibble(TXT)
```

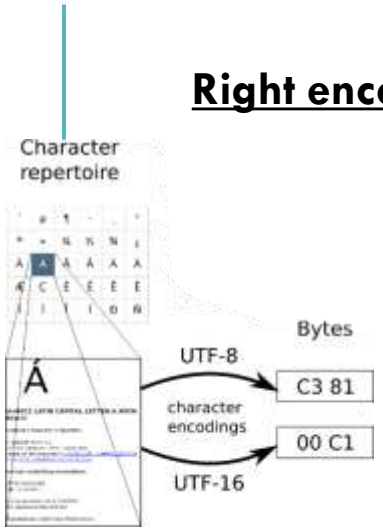


Mention three things that you learned.

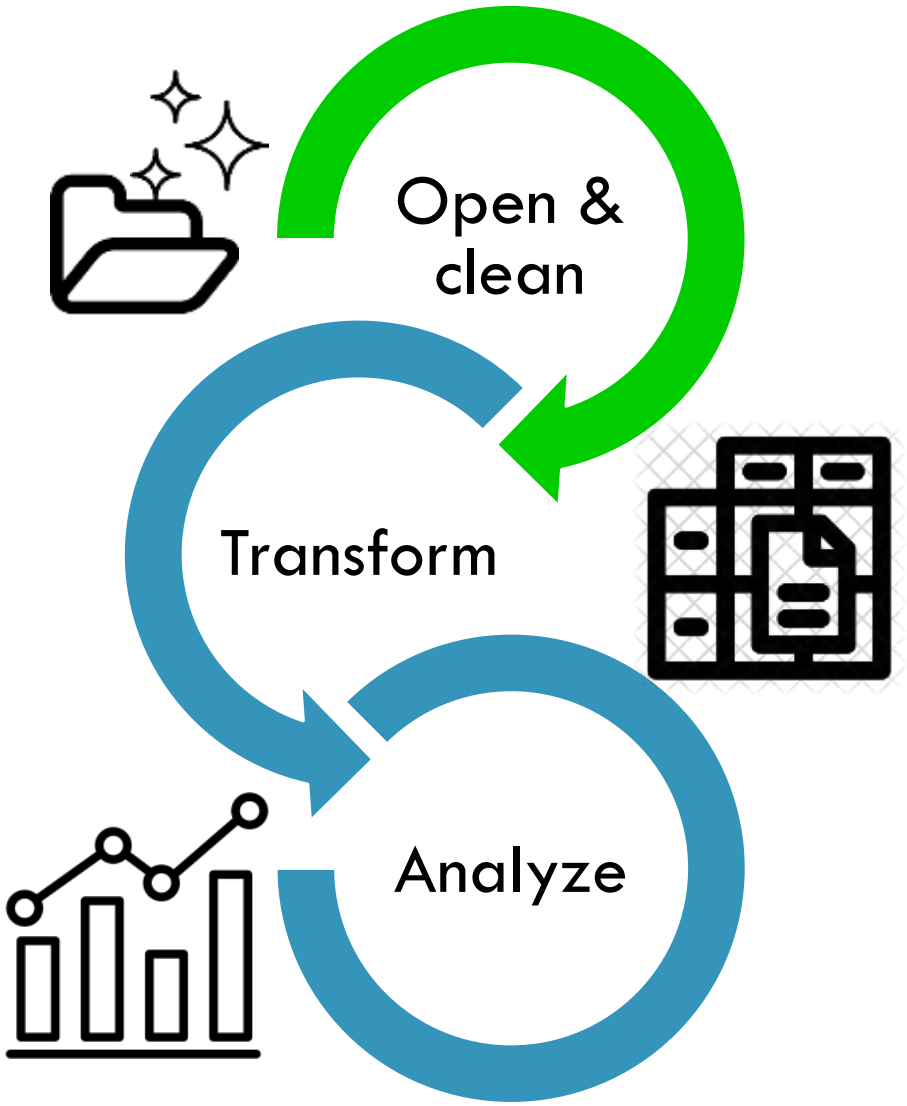
The pipeline

Mention three things that you learned.

Right encoding



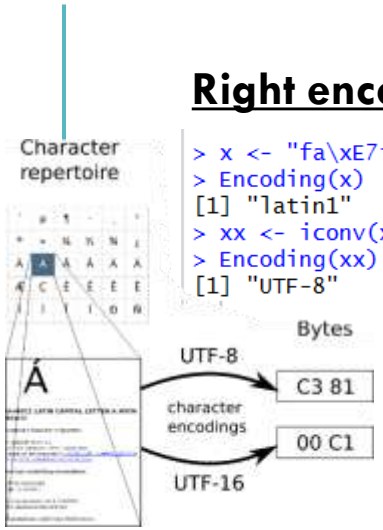

```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

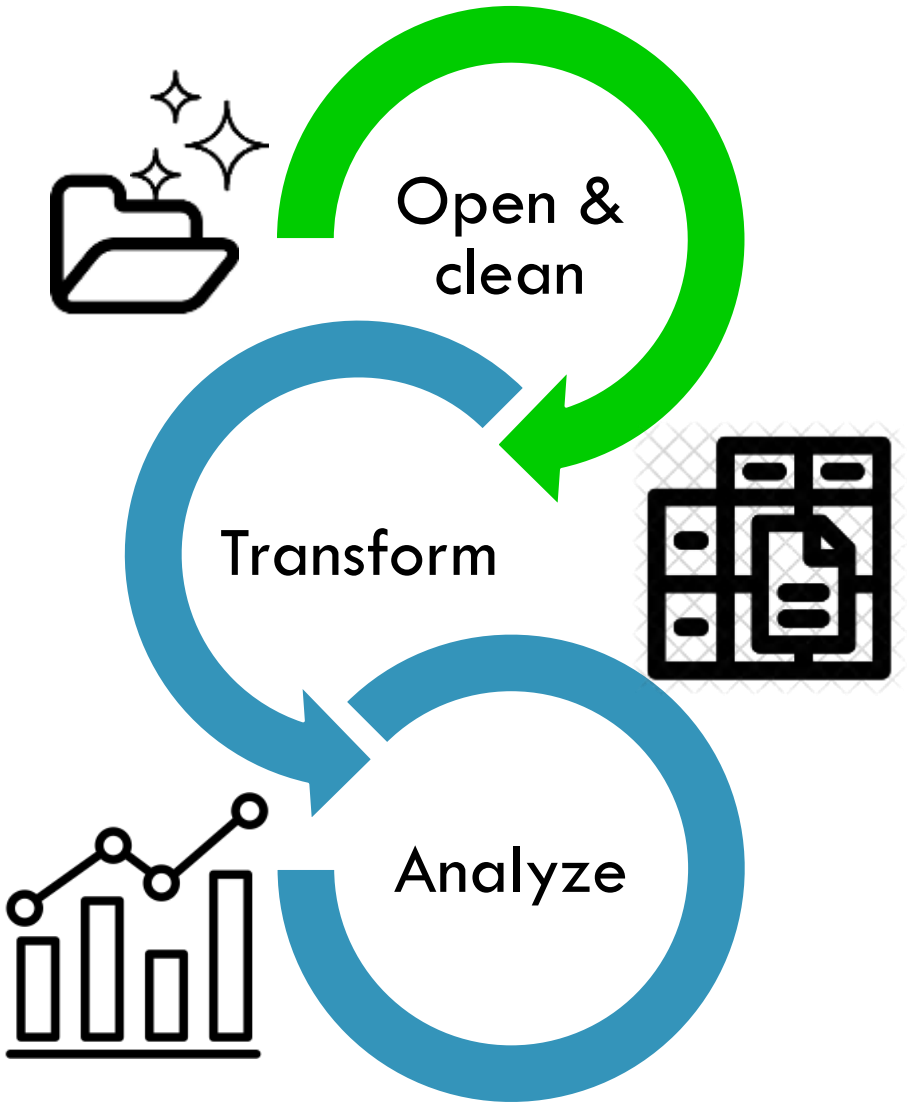
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

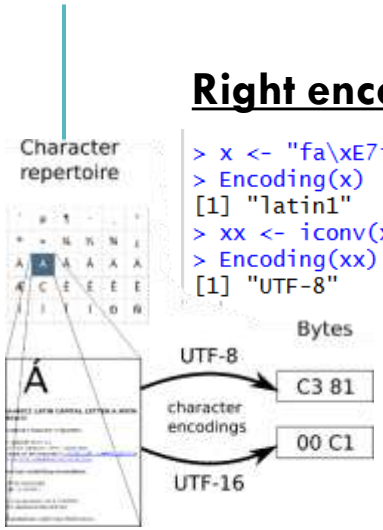
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

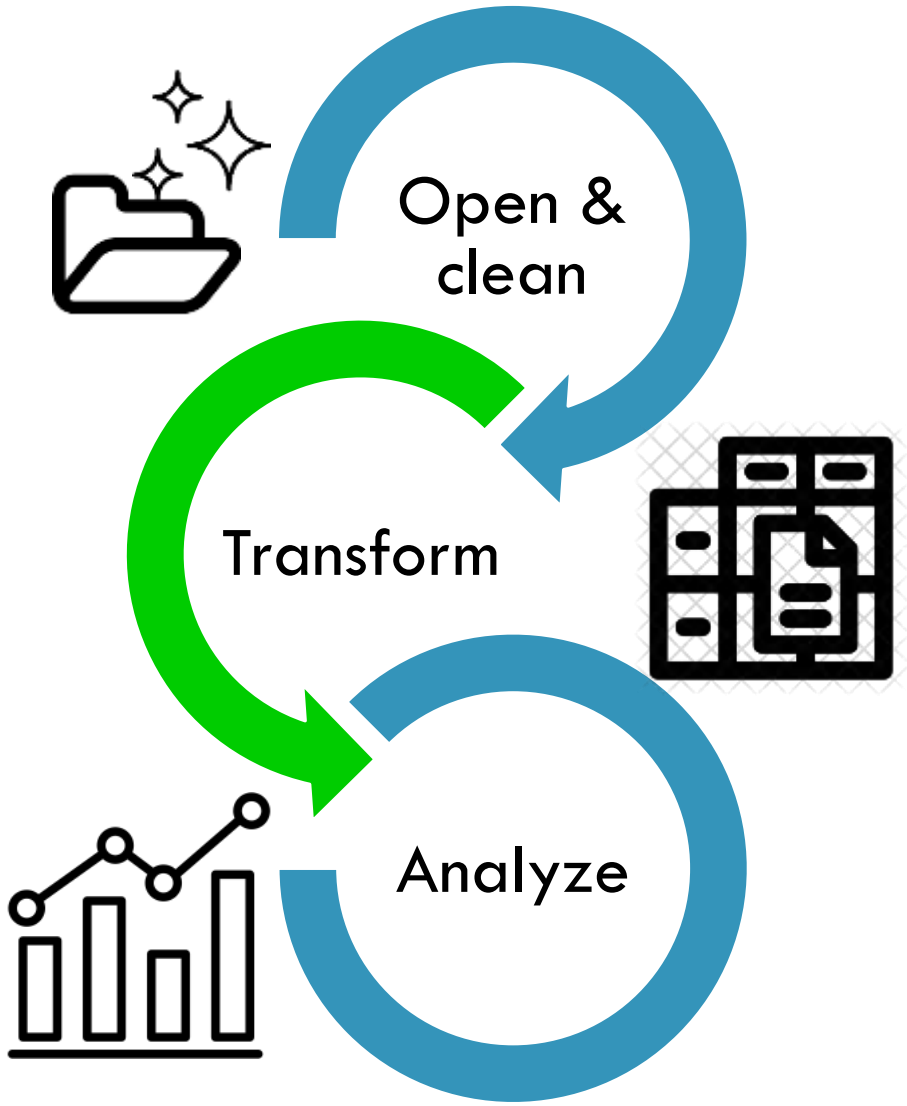
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

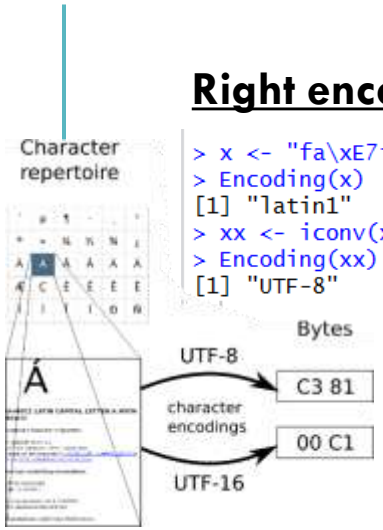
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

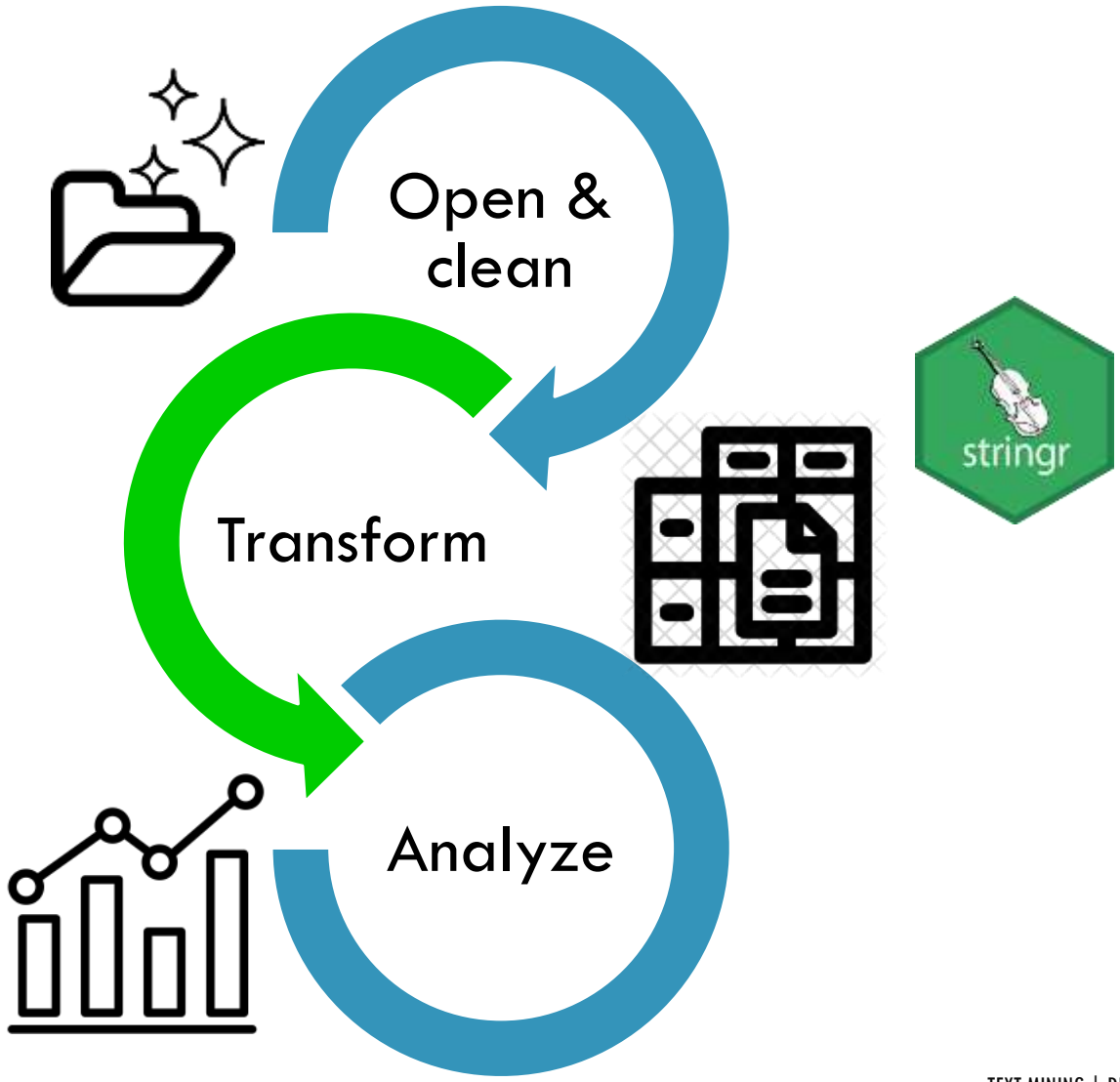
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

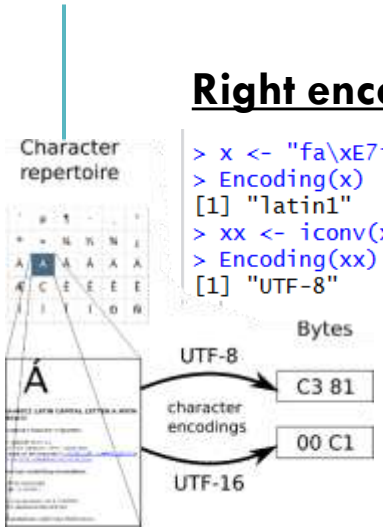
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

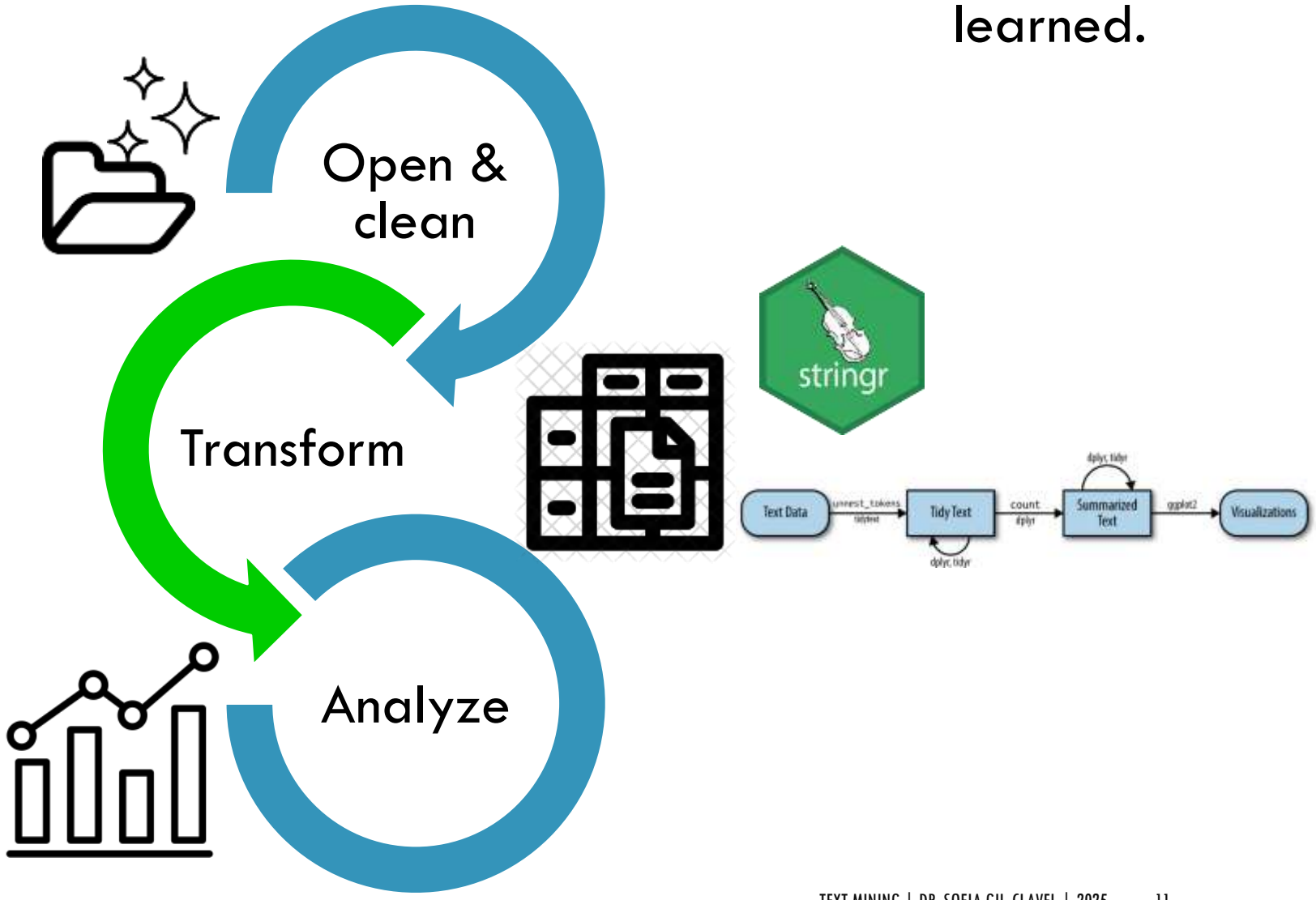
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

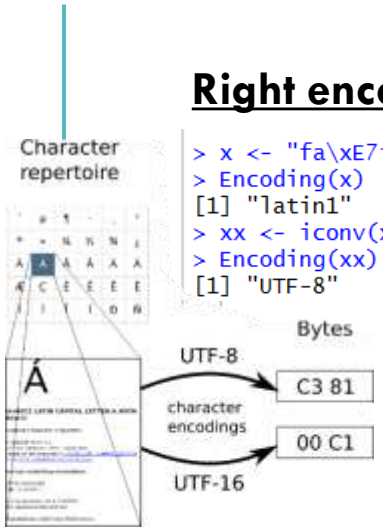
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

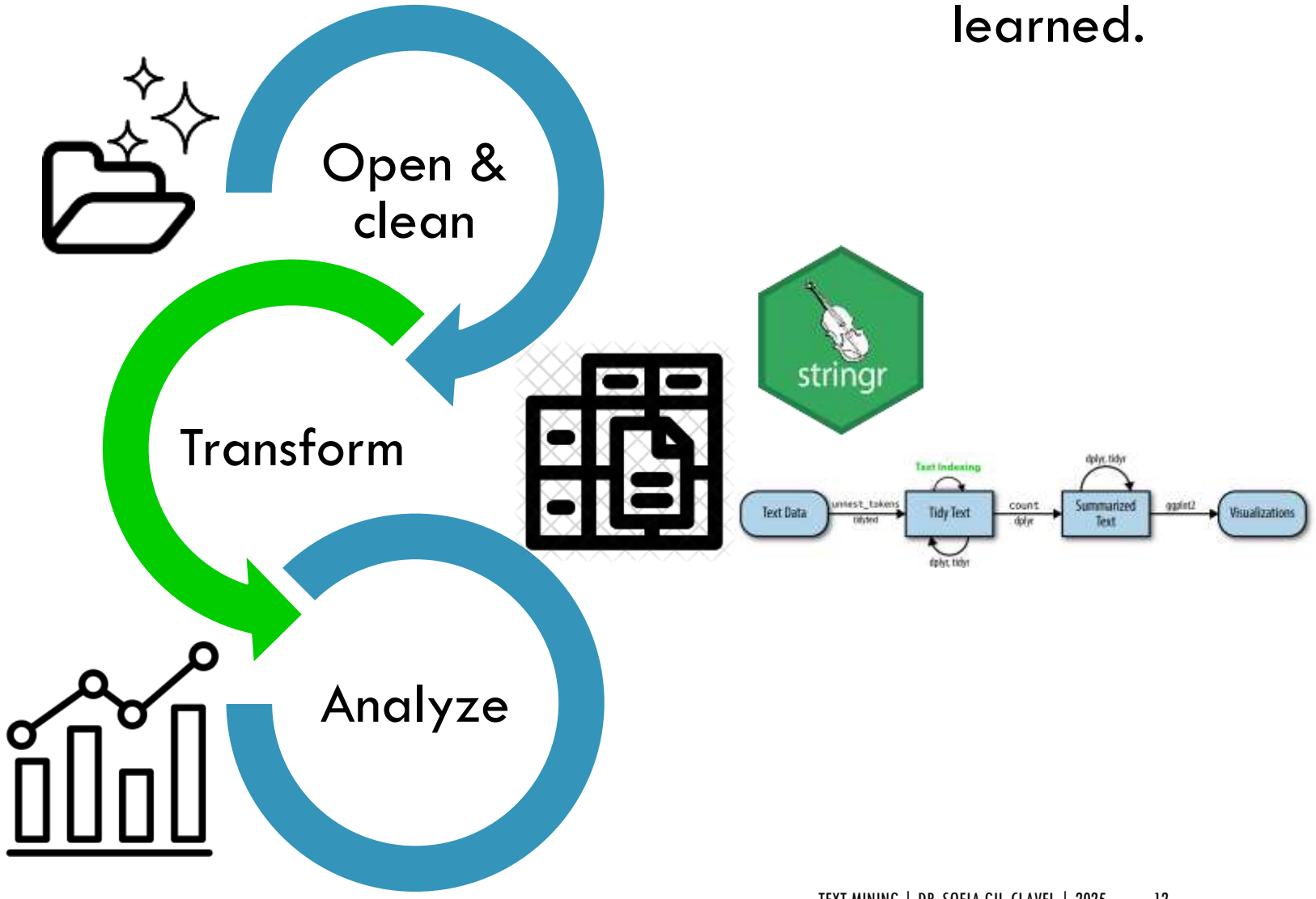
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

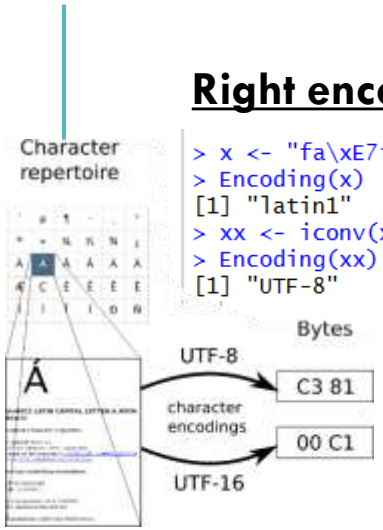
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

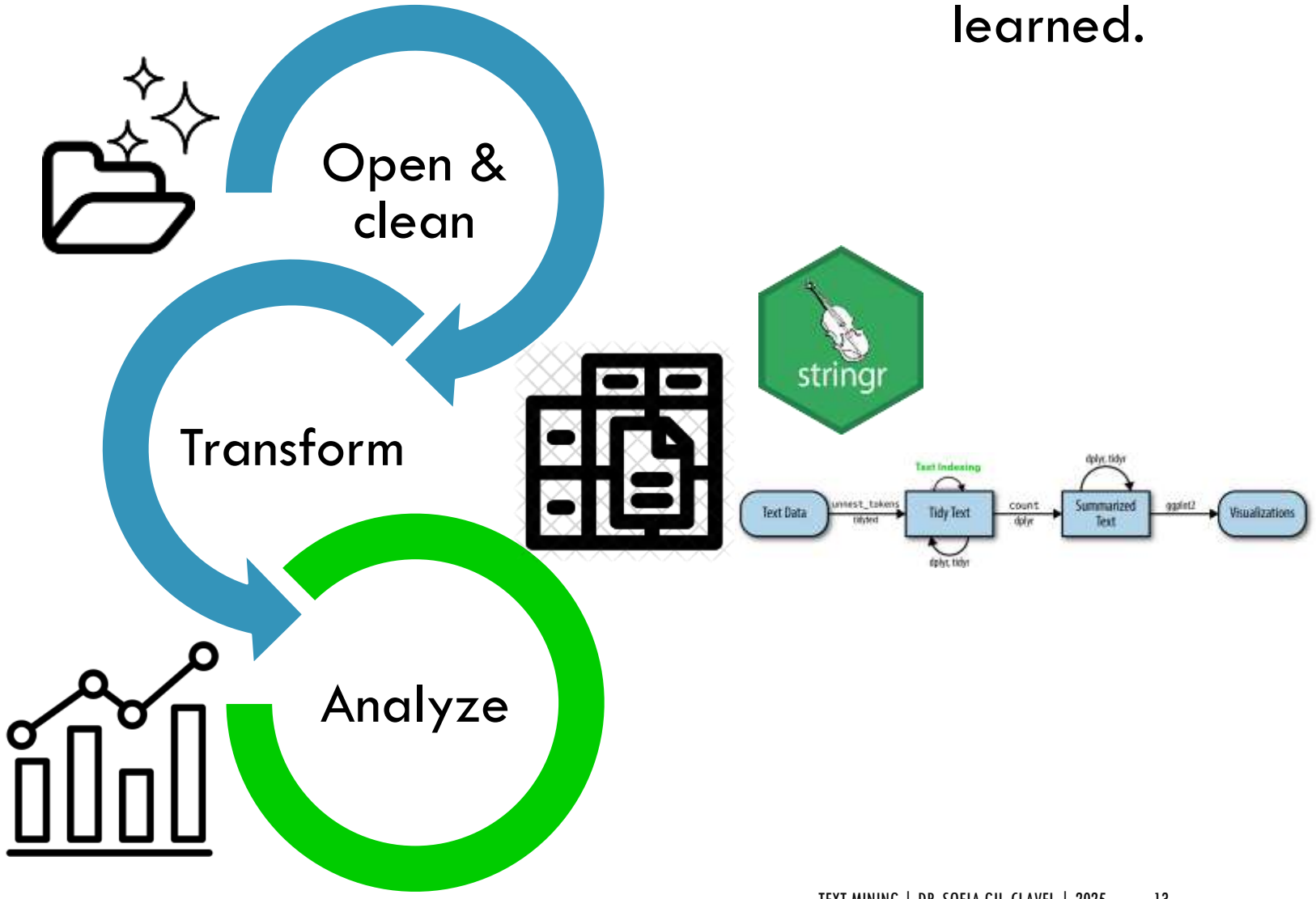
Mention three things that you learned.

Right encoding




```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

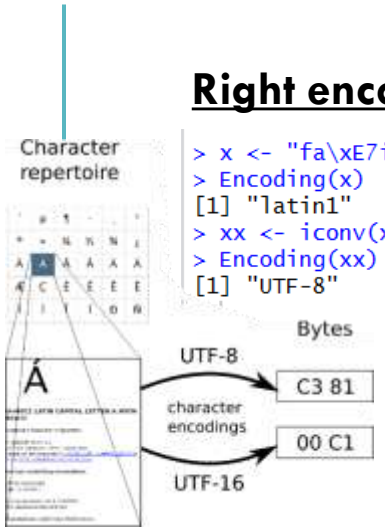
```
TXT=read.delim(...)
TXT=tibble(TXT)
```



The pipeline

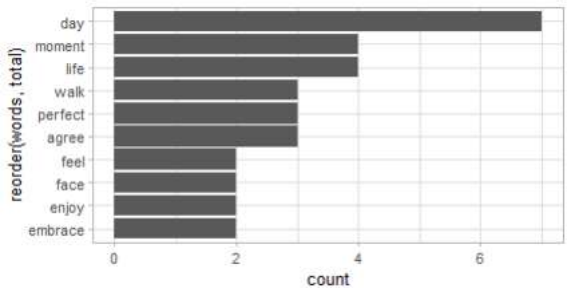
Mention three things that you learned.

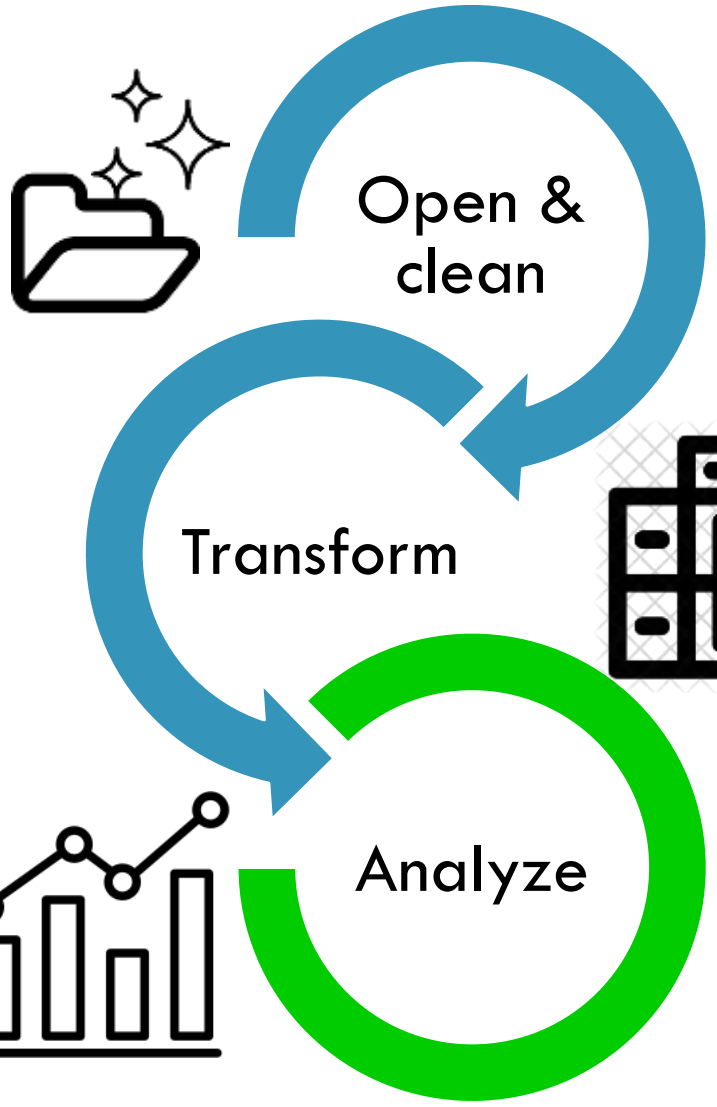
Right encoding

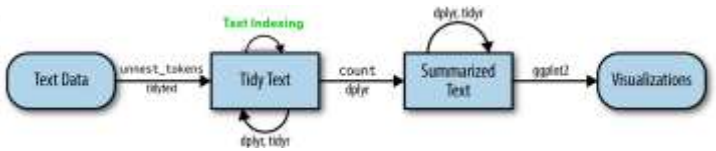



```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

```
TXT=read.delim(...)
TXT=tibble(TXT)
```



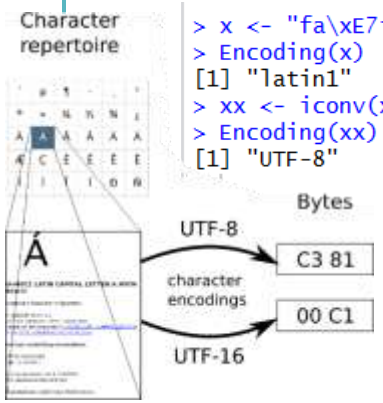




The pipeline

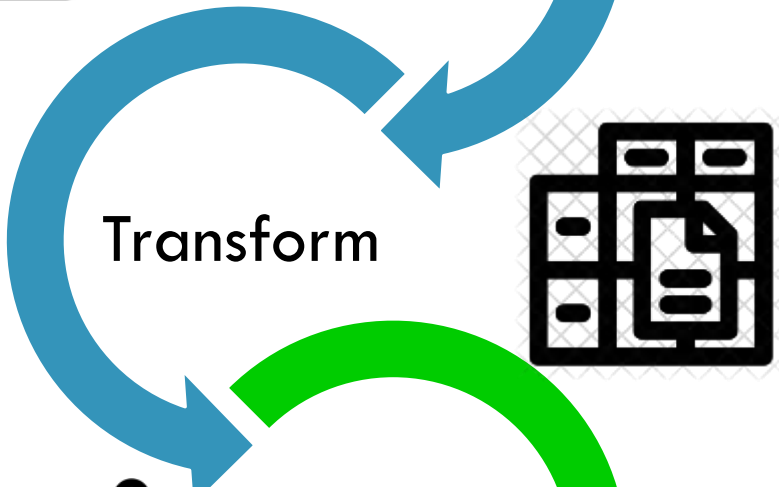
Mention three things that you learned.

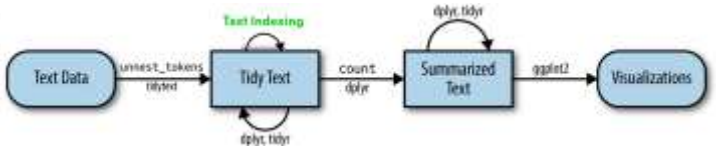
Right encoding

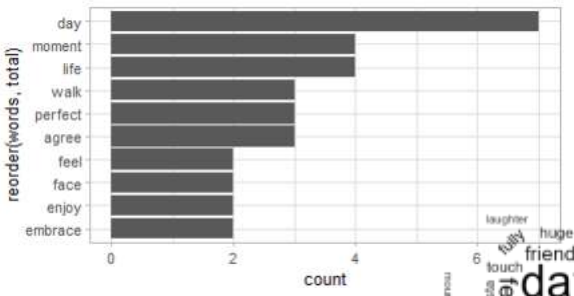



```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

```
TXT=read.delim(...)
TXT=tibble(TXT)
```



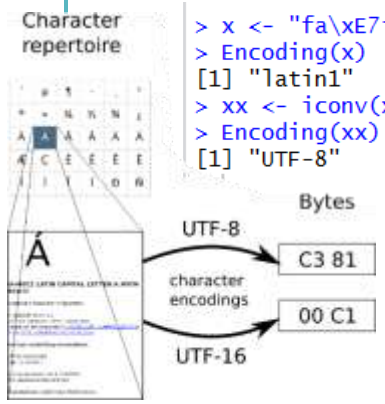




The pipeline

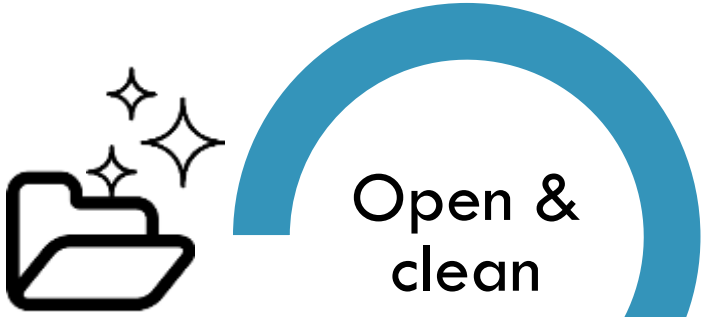
Mention three things that you learned.

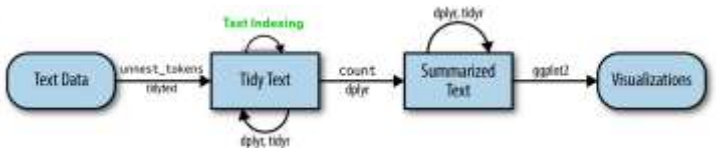
Right encoding

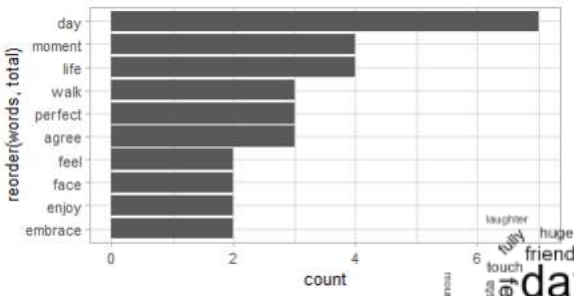



```
> x <- "fa\xE7ile"
> Encoding(x)
[1] "latin1"
> xx <- iconv(x, "latin1", "UTF-8")
> Encoding(xx)
[1] "UTF-8"
```

```
TXT=read.delim(...)
TXT=tibble(TXT)
```

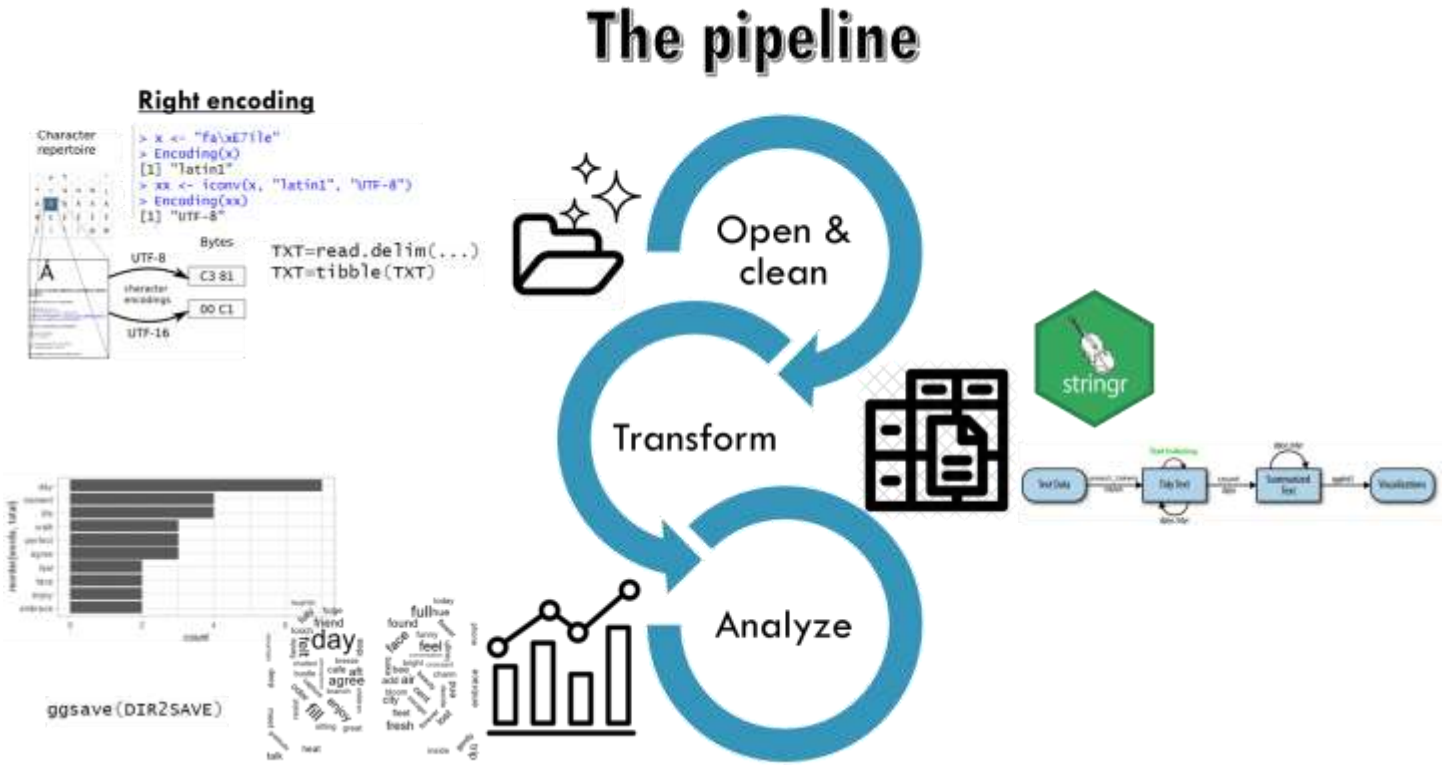




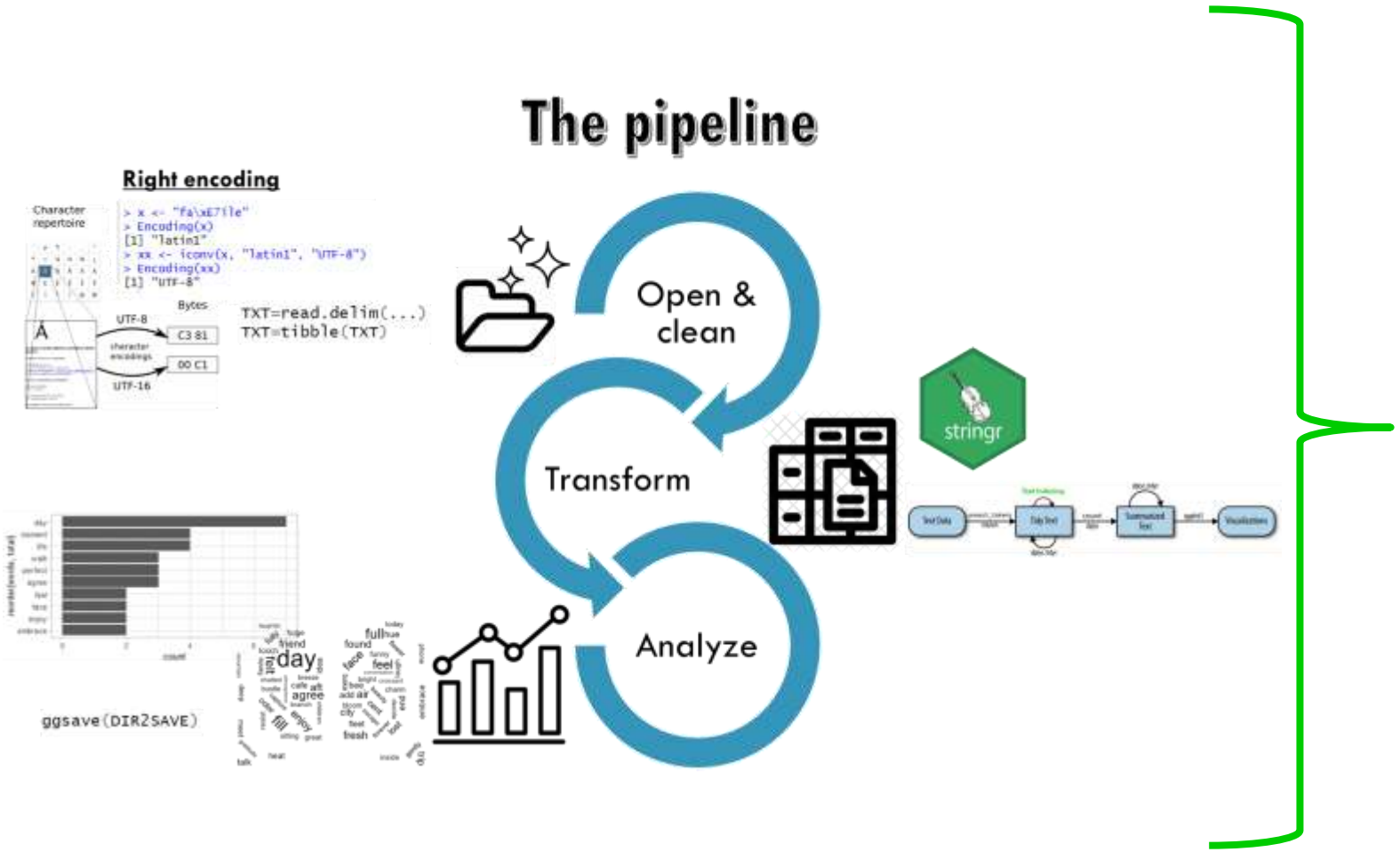



```
ggsave(DIR2SAVE)
```

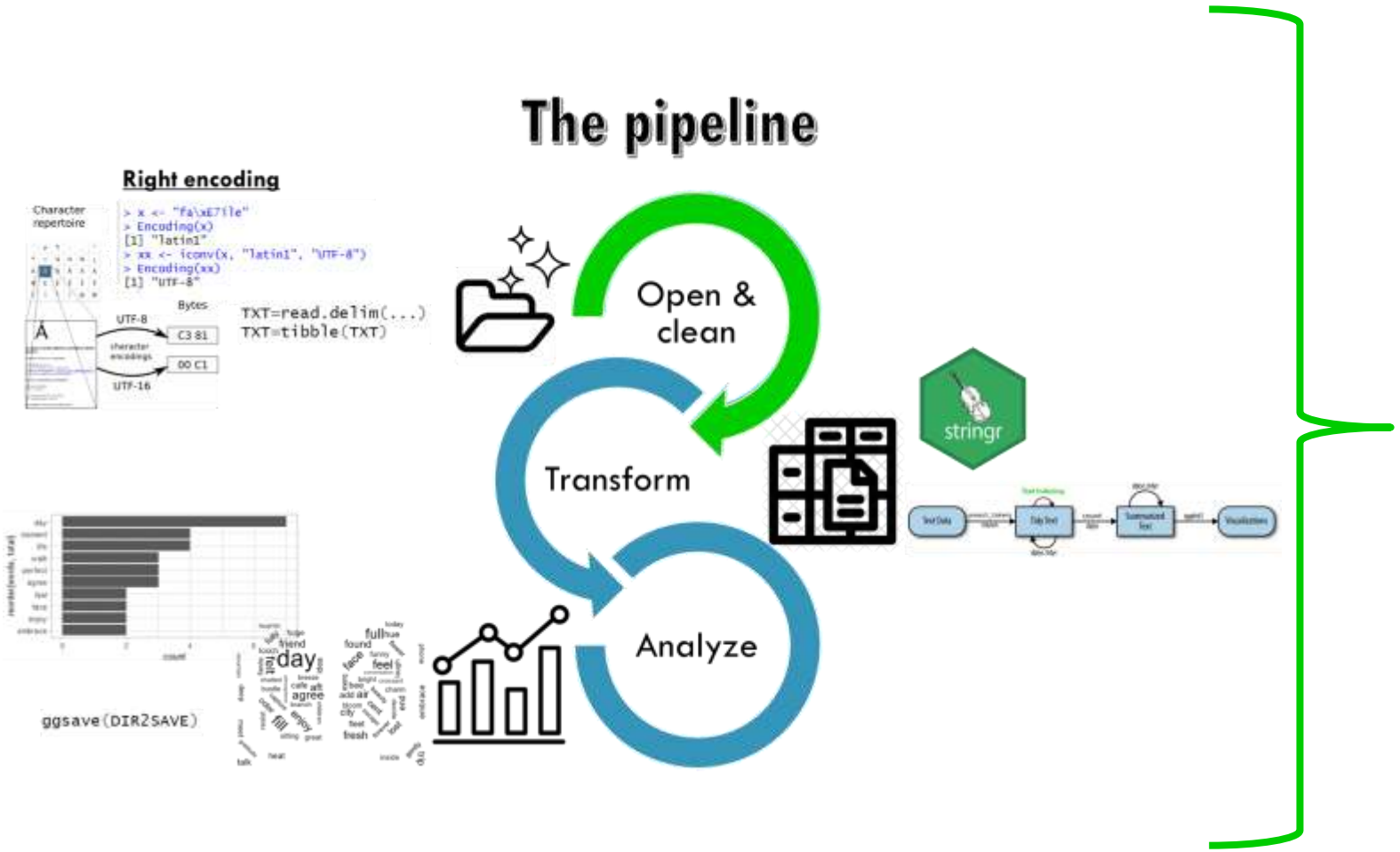
TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



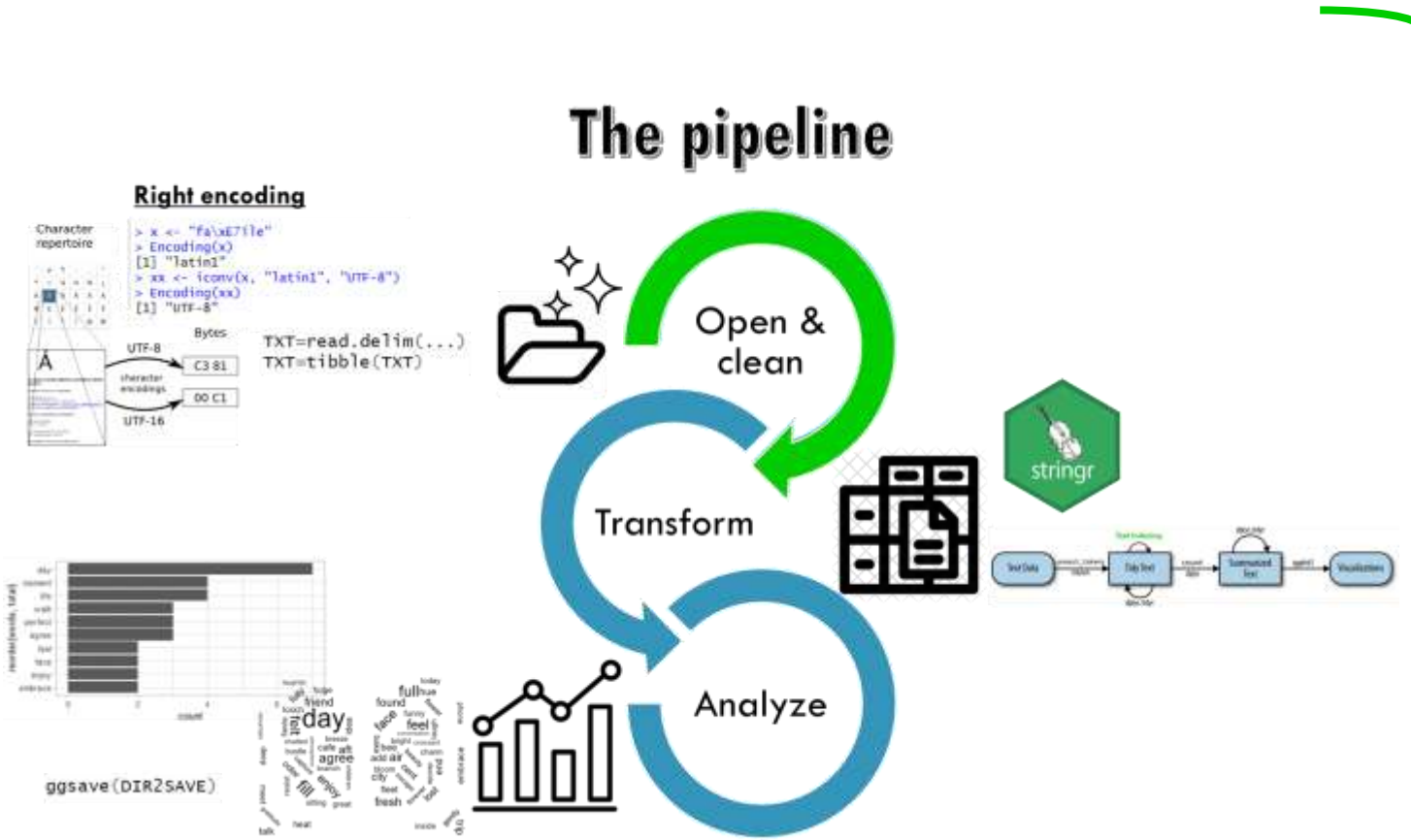
TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!

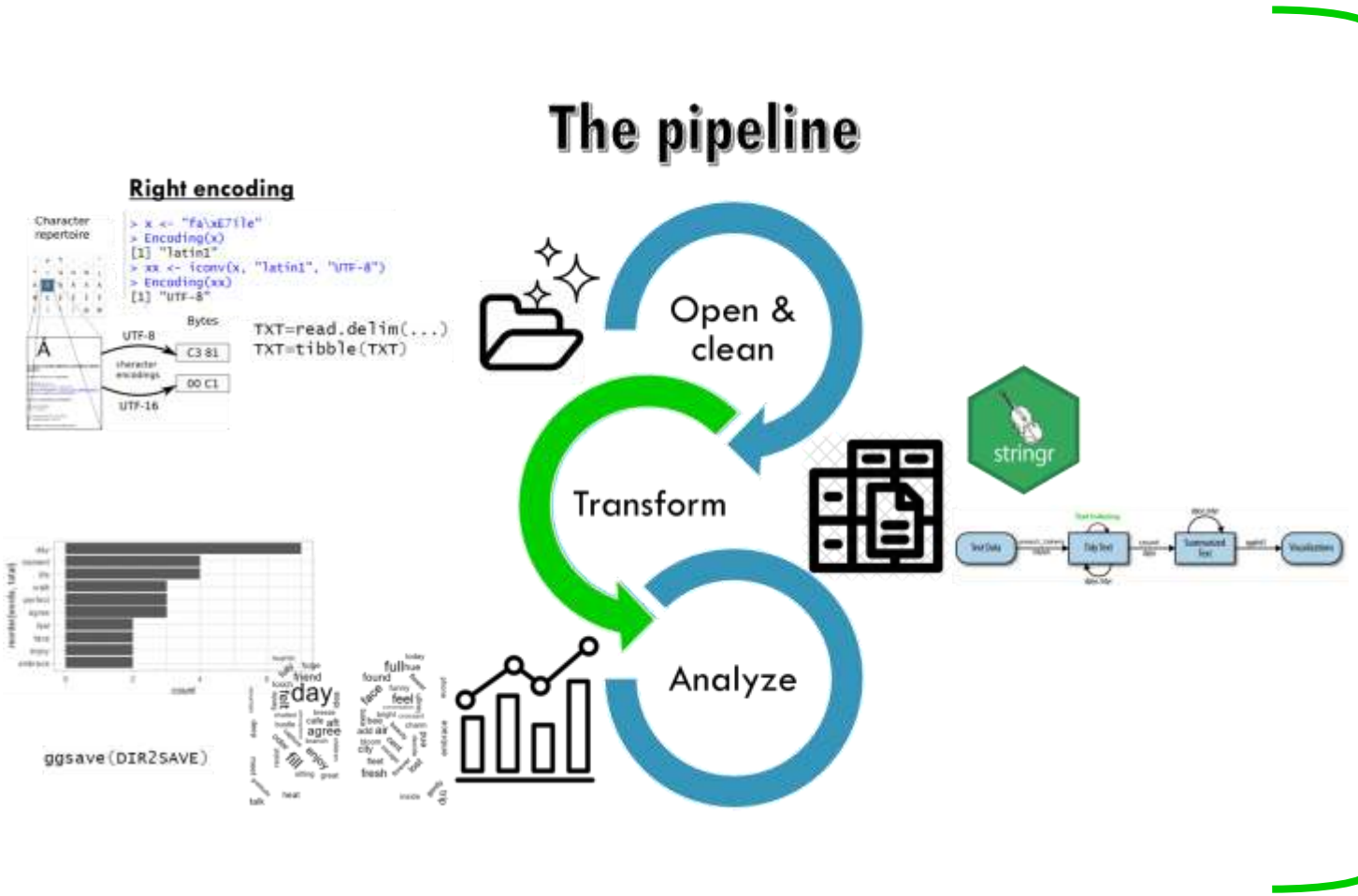


TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



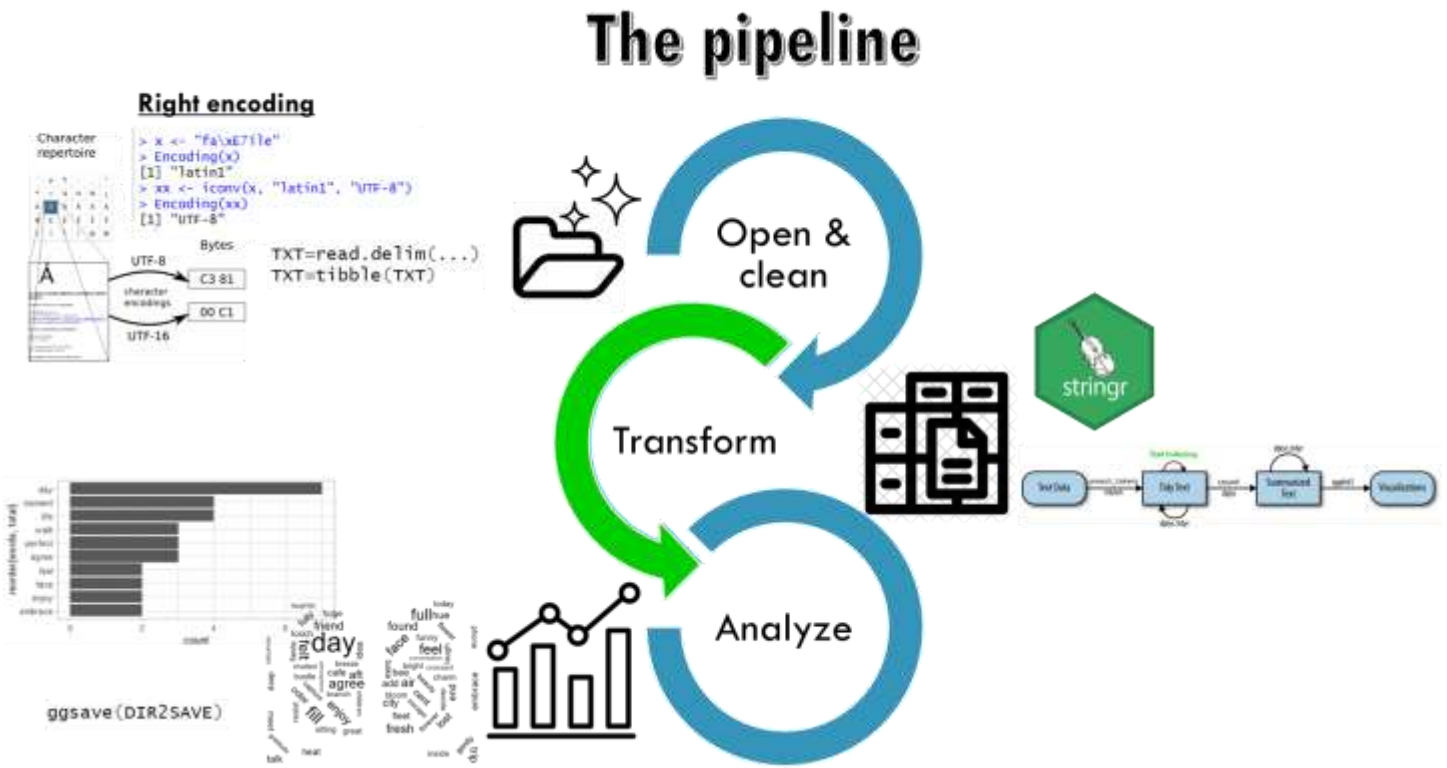
- **Move from 1 text to multiple texts.**

TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



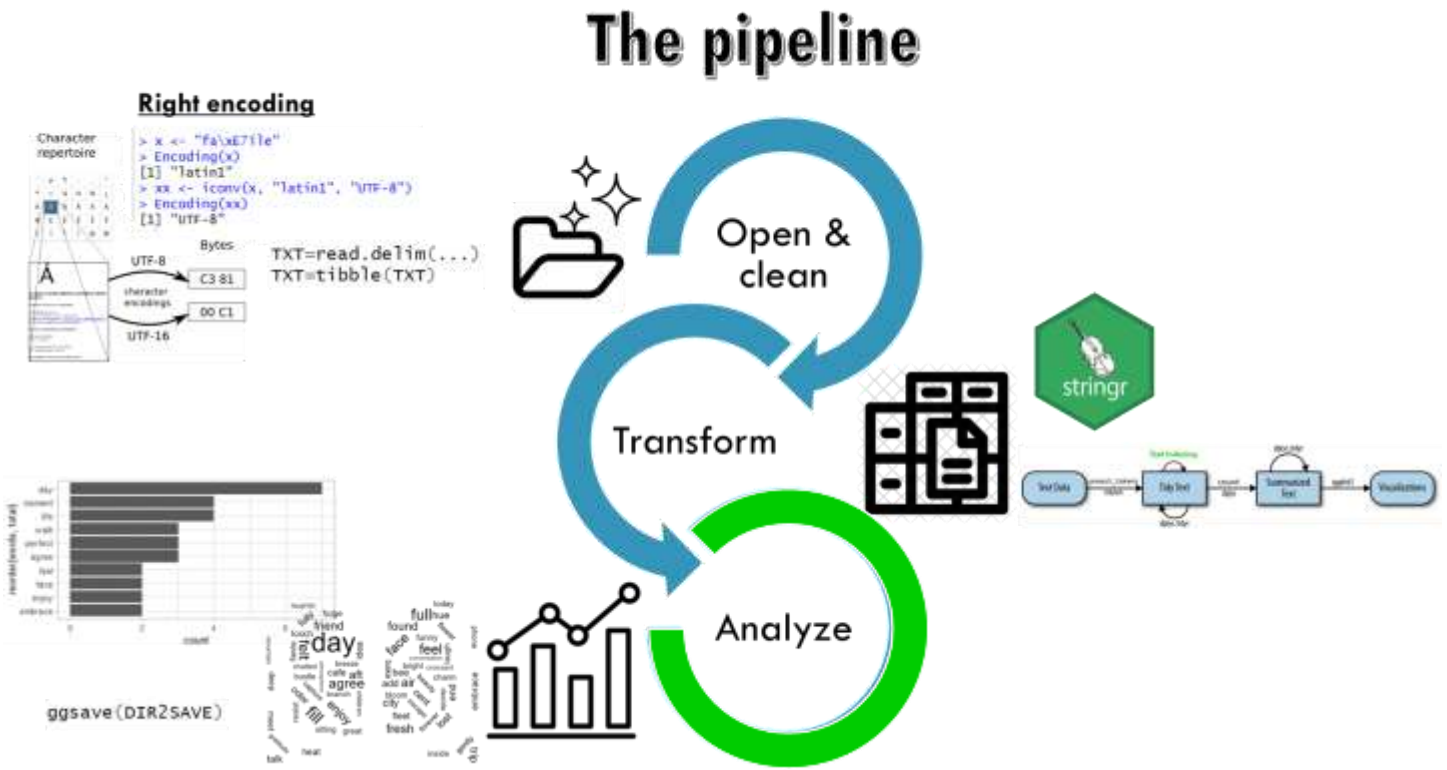
- Move from 1 text to multiple texts.
- **Move from 1 word to n-grams.**

TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



- Move from 1 text to multiple texts.
- Move from 1 word to n-grams.
- **Move from frequencies to measures that account for words that happen too often but do not mean anything.**

TODAY WE WILL CONTINUE APPLYING THIS PIPELINE!



- Move from 1 text to multiple texts.
- Move from 1 word to n-grams.
- Move from frequencies to measures that account for words that happen too often but do not mean anything.
- **Move from frequency plots to relational graphs (networks).**

2. TEXT MINING & DATAVIZ

1. What is text mining?
2. Word and document frequency
3. Relationships between words

2.1 WHAT IS TEXT MINING?

2.1 WHAT IS TEXT MINING?

Text mining, also known as text data mining, is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights.

JANE AUSTEN BOOKS

Source: https://www.amazon.nl/-/en/Jane-Austen/dp/1785995103

JANE AUSTEN BOOKS

What would be interesting to study using text mining?

Source: https://www.amazon.nl/-/en/Jane-Austen/dp/1785995103

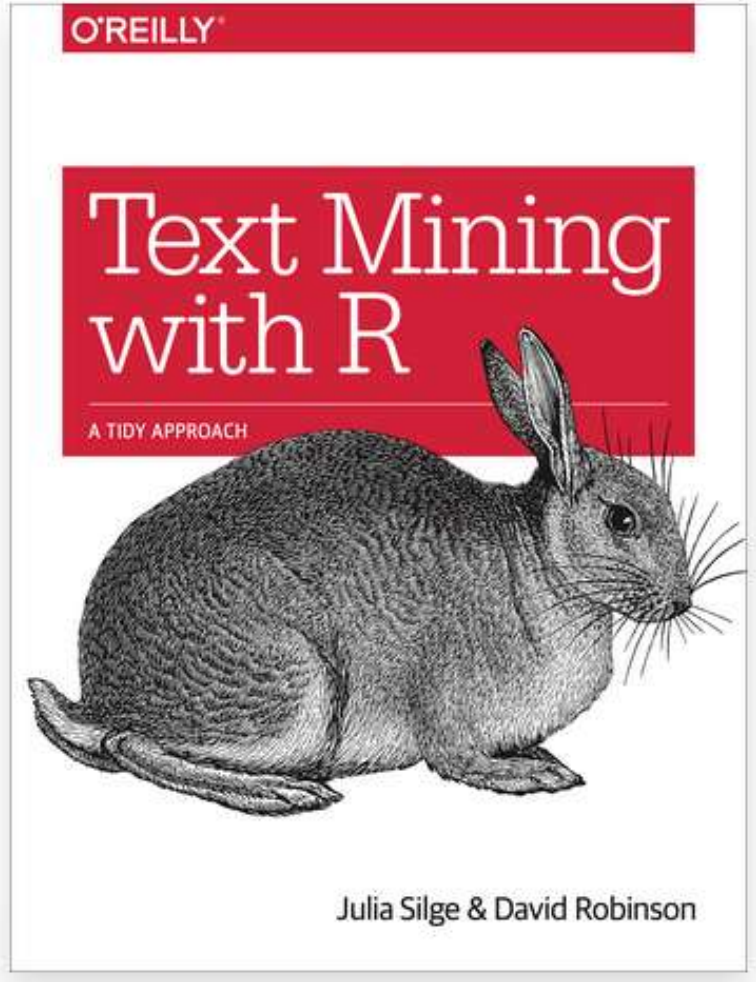
JANE AUSTEN BOOKS

What would be interesting to study using text mining?

Source: https://www.amazon.nl/-/en/Jane-Austen/dp/1785995103

2.2 WORD AND DOCUMENT FREQUENCY

❖ TIDYTEXT: HTTPS://WWW.TIDYTEXTMINING.COM/



We developed the tidytext (Silge and Robinson 2016) R package because we were familiar with many methods for data wrangling and visualization, but couldn't easily apply these same methods to text. We found that using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Treating text as data frames of individual words allows us to manipulate, summarize, and visualize the characteristics of text easily and integrate natural language processing into effective workflows we were already using.

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

Some measures of how important a word may be are:

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

Some measures of how important a word may be are:

- The ***term frequency*** **(tf),** how frequently a word occurs in a document.

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

Some measures of how important a word may be are:

- The ***term frequency*** **(tf),** how frequently a word occurs in a document.
- The term's **inverse document frequency (idf),** which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents.

$$idf(\text{term}) = \ln\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

Some measures of how important a word may be are:

- The ***term frequency*** **(tf),** how frequently a word occurs in a document.
- The term's **inverse document frequency (idf),** which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents.
- This can be **combined** with term frequency to **calculate a term's tf-idf** (the two quantities multiplied together), the frequency of a term adjusted for how rarely it is used.

2.2 WORD AND DOCUMENT FREQUENCY

A central question in text mining and natural language processing is **how to quantify what a document is about.**

Can we do this by looking at the words that make up the document?

Some measures of how important a word may be are:

- The ***term frequency*** **(tf),** how frequently a word occurs in a document.
- The term's **inverse document frequency (idf),** which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents.
- This can be **combined** with term frequency to **calculate a term's tf-idf** (the two quantities multiplied together), the frequency of a term adjusted for how rarely it is used.

2.2 TF-IDF

The idea of tf-idf is to find the important words for the content of each document by decreasing the weight for commonly used words and increasing the weight for words that are not used very much in a collection or corpus of documents, in this case, the group of Jane Austen's novels as a whole. Calculating tf-idf attempts to find the words that are important (i.e., common) in a text, but not too common.

We will use the function from tidytext:

```
bind_tf_idf(word, book, n)
```

10 MINS BREAK

2.3 RELATIONSHIPS BETWEEN WORDS

2.3 RELATIONSHIPS BETWEEN WORDS

Many interesting text analyses are based on the relationships between words, whether examining which words tend to follow others immediately, or that tend to co-occur within the same documents.

N-GRAMS

We can also tokenize into consecutive sequences of words, called n-grams. By seeing how often word X is followed by word Y, we can then build a model of the relationships between them.

N-GRAMS: TF-IDF

A bigram can also be treated as a term in a document in the same way that we treated individual words. For example, we can look at the tf-idf of bigrams across Austen novels. These tf-idf values can be visualized within each book, just as we did for words.

CORRELATING PAIRS OF WORDS

We may instead want to examine correlation among words, which indicates how often they appear together relative to how often they appear separately.

We'll focus on the phi coefficient, a common measure for binary correlation. The focus of the phi Φ coefficient is how much more likely it is that either both word X and Y appear, or neither do, than that one appears without the other.

	$y = 1$	$y = 0$	total
$x = 1$	n_{11}	n_{10}	$n_{1\bullet}$
$x = 0$	n_{01}	n_{00}	$n_{0\bullet}$
total	$n_{\bullet 1}$	$n_{\bullet 0}$	n

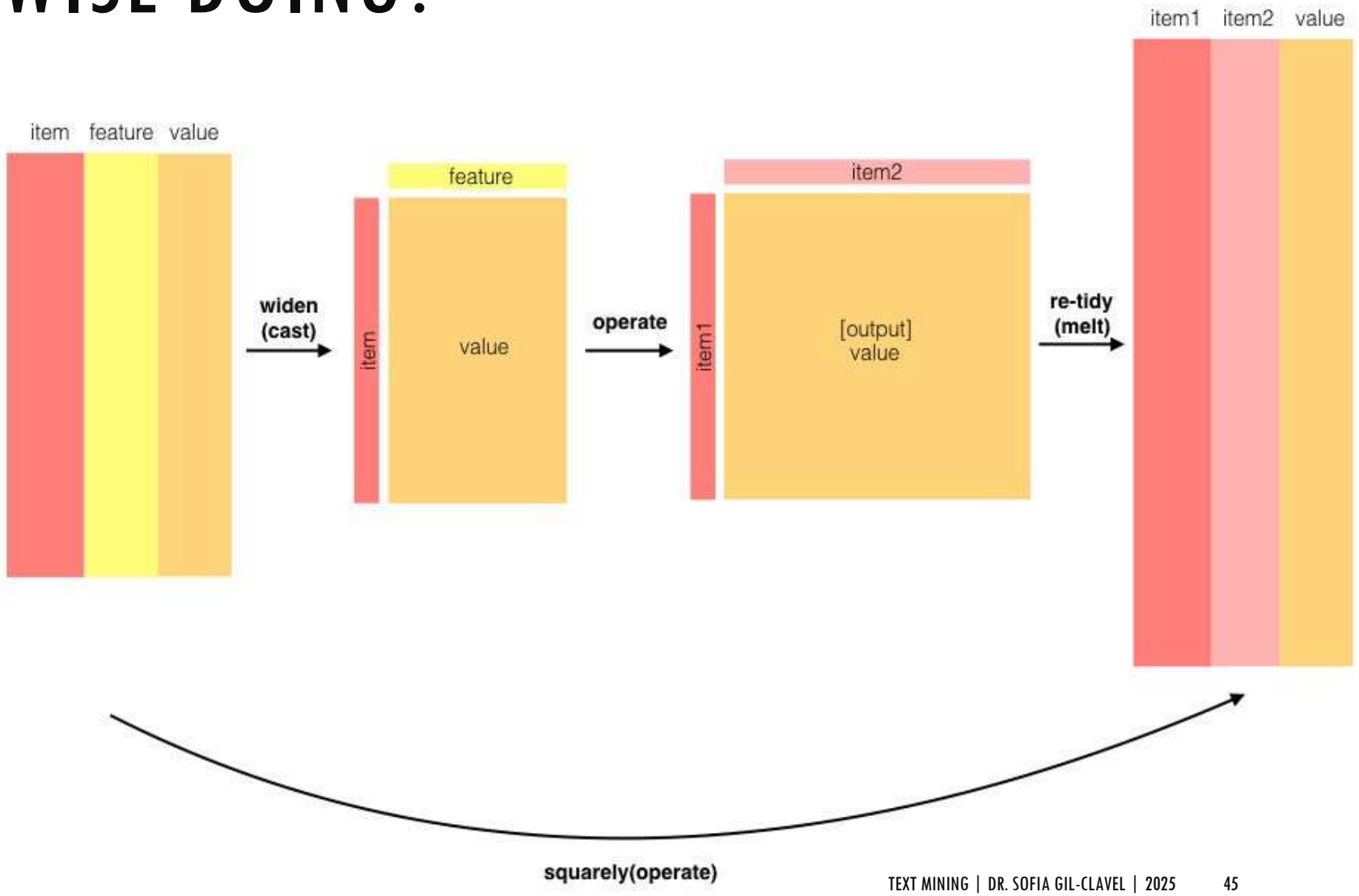
$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\bullet}n_{0\bullet}n_{\bullet 0}n_{\bullet 1}}}$$

This is done with the package widyr and its function:

```
pairwise_cor(word, section, sort = TRUE)
```

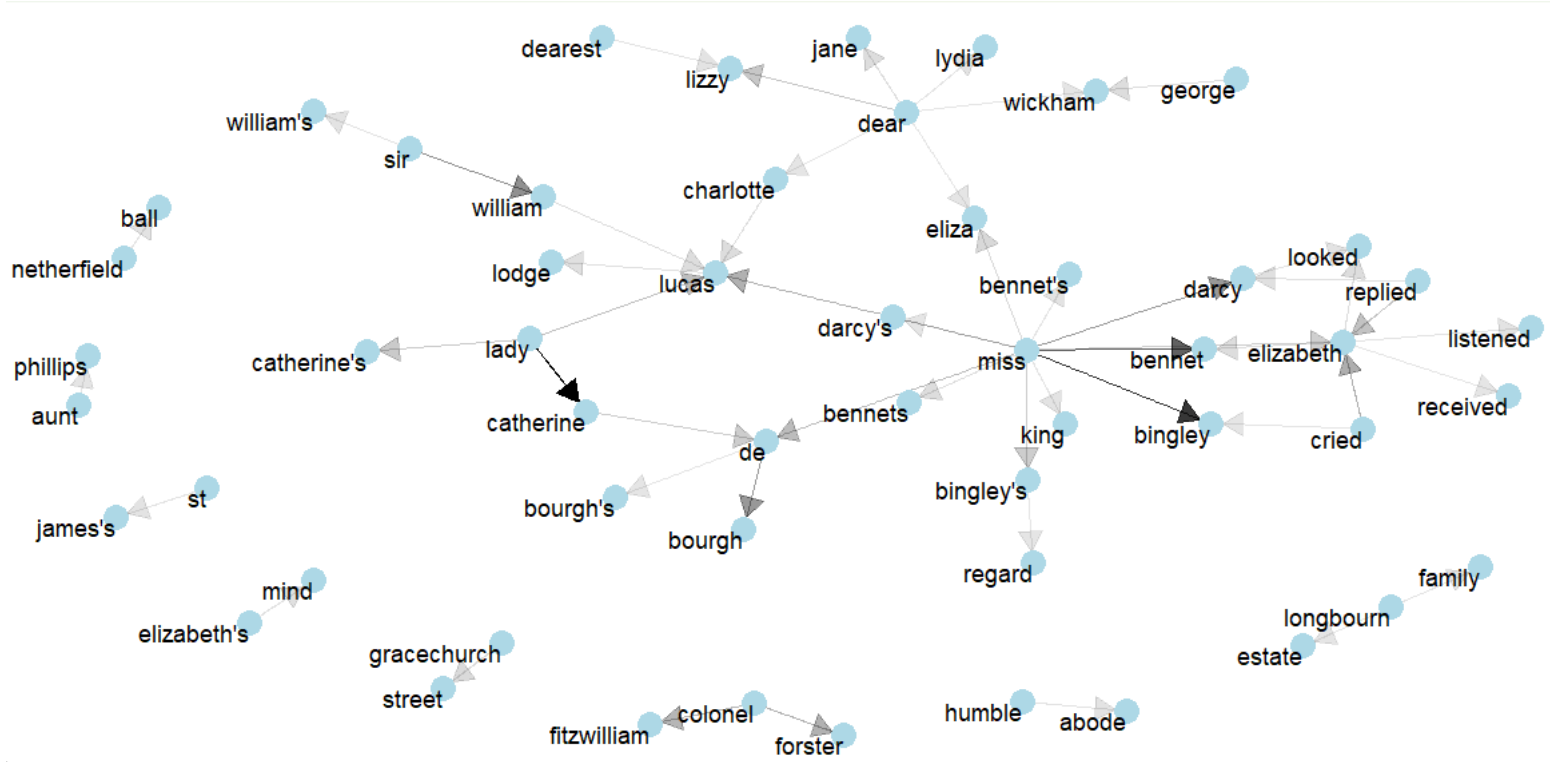
WHAT IS PAIRWISE DOING?

Counts the number of times each pair of *items* appear together within a group defined by "*feature*". In this case, it counts the number of times each pair of words appear together within a section. Later it uses this information to calculate the correlation.



NETWORK OF N-GRAMS RELATIONS

We can visualize the n-grams using the different metrics: counts, tf-idf, and correlations.



3. QUICK OVERVIEW OF ADVANCE TOPICS

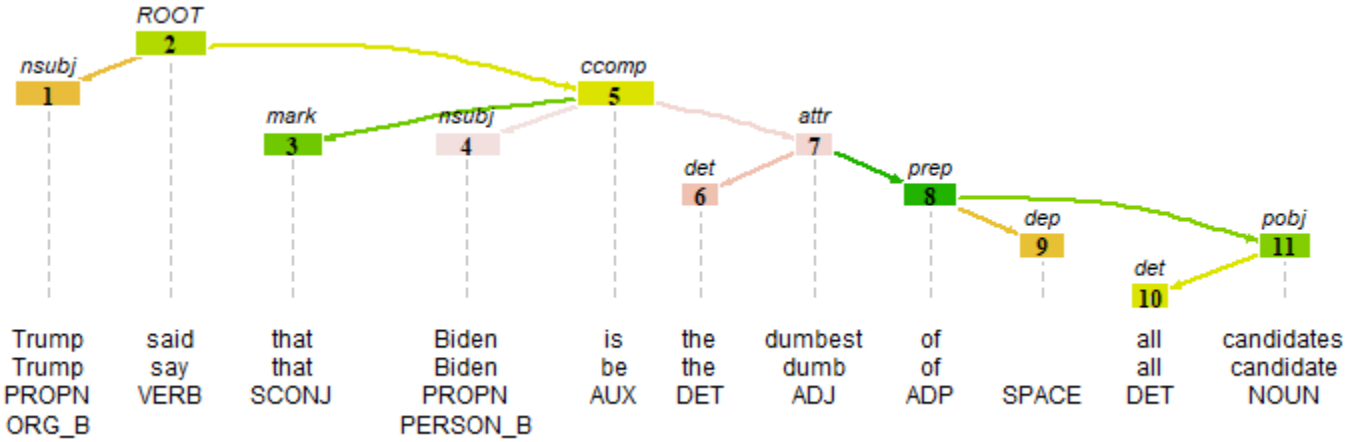
1. Natural Language Processing (NLP)
2. Parts-Of-Speech (POS)
3. Visualizing Trees

3.1 NATURAL LANGUAGE PROCESSING (NLP)

NLP enables computers and digital devices to recognize, understand, and generate text and speech by combining computational linguistics—the rule-based modeling of human language—together with statistical modeling, machine learning, and deep learning.

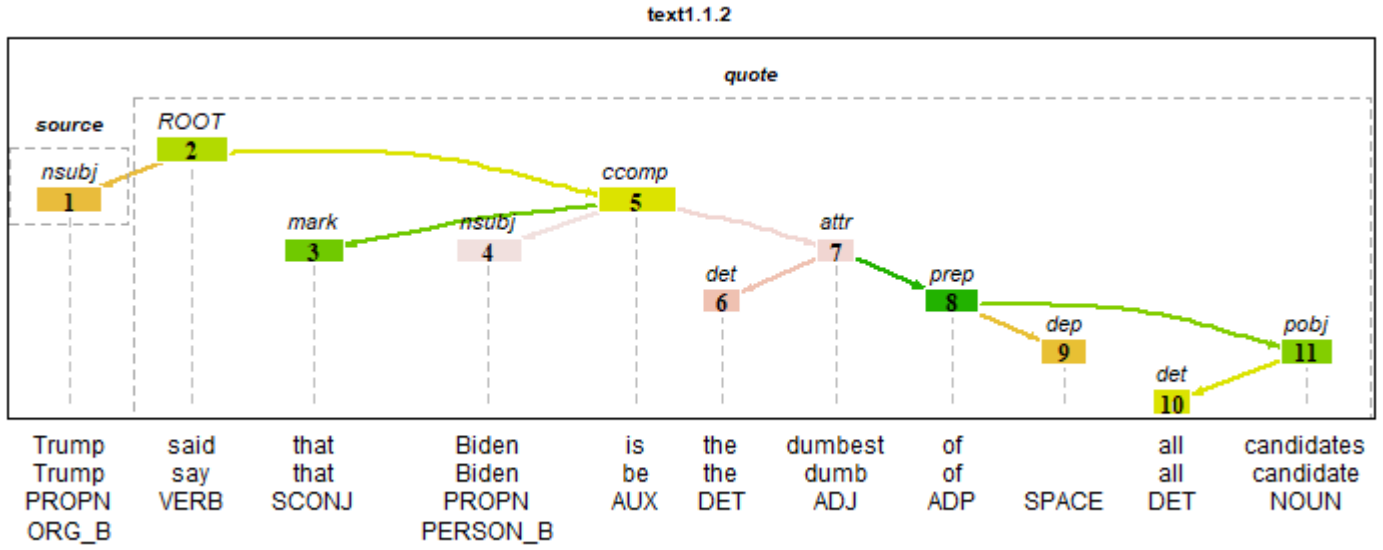
3.2 PART-OF-SPEECH (POS)

Part of speech: a class of words (as adjectives, adverbs, conjunctions, interjections, nouns, prepositions, pronouns, or verbs) identified according to the kinds of ideas they express and the way they work in a sentence.



Source: Welbers, Kasper, Wouter Van Atteveldt, and Jan Kleinnijenhuis. "Extracting Semantic Relations Using Syntax: An R Package for Querying and Reshaping Dependency Trees." *Computational Communication Research* 3, no. 2 (October 1, 2021): 1–16. https://doi.org/10.5117/CCR2021.2.003.WELB.

WHO DOES WHAT TO WHOM AND ACCORDING TO WHAT SOURCE



Source: Welbers, Kasper, Wouter Van Atteveldt, and Jan Kleinnijenhuis. "Extracting Semantic Relations Using Syntax: An R Package for Querying and Reshaping Dependency Trees." *Computational Communication Research* 3, no. 2 (October 1, 2021): 1–16. https://doi.org/10.5117/CCR2021.2.003.WELB.

Join us!

https://forms.office.com/e/8Bgd2YsasJ

All about the lab:

https://societal-analytics.nl/

Contact us at:

analytics-lab.fsw@vu.nl.

THANKS!

Dr. Sofia Gil-Clavel

https://sofiag1l.github.io/