

# Lecture 21: Strings and regular expressions

# Recap: regular expressions

A *regular expression* is a pattern used to find matches in text.

**Example:** suppose I want to extract just the lecture number from the following file name. How would I do that?

```
1 "teaching/sta279-f23/slides/lecture_22.qmd"
```

Find number that comes after —

$(?, L = \_ ) \Rightarrow d +$

# Recap: regular expressions

A *regular expression* is a pattern used to find matches in text.

**Example:** suppose I want to extract just the lecture number from the following file name. How would I do that?

```
1 str_extract("teaching/sta279-f23/slides/lecture_22.qmd", "\\d+")
```

```
[1] "279"
```

```
1 str_extract("teaching/sta279-f23/slides/lecture_22.qmd", "_\\d+")
```

```
[1] "_22"
```

```
1 str_extract("teaching/sta279-f23/slides/lecture_22.qmd",  
2              "(?<=_)\\d+")
```

```
[1] "22"
```

# Recap: regular expressions

Last time, we learned the following regular expression tools:

- `\d` matches any digit (in R, have to type `\\d` because we write the regex in a string)
- `.` matches any character (except `\n`)
- `+` means “at least once”
- `(?<=)` and `(?=)` are positive lookbehinds and lookaheads
- `|` is alternation (one pattern or another)

# Recap: tools for working with strings

So far, we have learned the following:

- `str_extract` extracts the first match

```
1 str_extract("teaching/sta279-f23/slides/lecture_22.qmd", "\\d+")  
[1] "279"
```

- `str_extract_all` extracts all matches

```
1 str_extract_all("teaching/sta279-f23/slides/lecture_22.qmd", "\\d+")  
[[1]]  
[1] "279" "23"  "22"
```

Goal for today: learn more string and regex tools!

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

I want to identify the files in the `research` folder. What pattern would I want to match?

`~research"`

or

`~ ^research"`

↑

`"starts with"`

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

I want to identify the files in the `research` folder. What pattern would I want to match?

```
1 str_detect(file_names, "research")
```

```
[1] TRUE TRUE FALSE FALSE
```

returns T or F for each entry in vector

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

I want to identify the files in the `research` folder. What pattern would I want to match?

```
1 str_subset(file_names, "research")
```

```
[1] "research/project1/code.R" "research/project1/data.csv"
```

return the  
strings  
which match  
the pattern



# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

I want to identify the files in the `research` folder. What pattern would I want to match?

```
1 str_view(file_names, "research")
```

```
[1] | <research>/project1/code.R  
[2] | <research>/project1/data.csv
```



part that  
matched the pattern

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

How would I select only the csv files?

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "teaching/sta279/lecture1.qmd",  
4                 "teaching/sta279/example_data.csv")
```

How would I select only the csv files?

```
1 str_subset(file_names, "csv")
```

```
[1] "research/project1/data.csv"  
"teaching/sta279/example_data.csv"
```

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "research/project2/sim_output.csv",  
4                 "teaching/sta279/lecture1.qmd",  
5                 "teaching/sta279/example_data.csv")
```

How would I select only the csv files in the research directory?

option 1:

```
str_subset(file_names, "research") >>  
str_subset("csv")
```

option 2: `str_subset(file_names, "research\\.+\\.csv")`

research          csv  
                  something



. +

(any character,  
one or more  
times)

# Some helpful string functions

**Example:** Suppose I have the following file names:

```
1 file_names <- c("research/project1/code.R",  
2                 "research/project1/data.csv",  
3                 "research/project2/sim_output.csv",  
4                 "teaching/sta279/lecture1.qmd",  
5                 "teaching/sta279/example_data.csv")
```

How would I select only the csv files in the research directory?

```
1 str_subset(file_names, "research.+csv")
```

```
[1] "research/project1/data.csv"  
"research/project2/sim_output.csv"
```

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select just raspberry and blackberry?

*-berry"*

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select just raspberry and blackberry?

```
1 str_view(strings, "berry")
```

```
[3] | rasp<berry>  
[4] | black<berry>
```

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select “raspberry”, “blackberry”, “grrreat”, and “random”?

contain r

"r"



# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select “raspberry”, “blackberry”, “grrreat”, and “random”?

```
1 str_view(strings, "r")
```

```
[3] | <r>aspbe<r><r>y  
[4] | blackbe<r><r>y  
[5] | g<r><r><r>eat  
[6] | <r>andom
```

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select just “raspberry”, “blackberry”, and “grrreat”?

rr

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```


How would I select just “raspberry”, “blackberry”, and “grrreat”?

 *at least 2 r's*

```
1 str_view(strings, "rr+")
```

```
[3] | raspbe<rr>y  
[4] | blackbe<rr>y  
[5] | g<rrr>eat
```

```
1 str_view(strings, "r{2,}")
```

 *at least 2 r's*

```
[3] | raspbe<rr>y  
[4] | blackbe<rr>y  
[5] | g<rrr>eat
```

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select just “grrreat”?

```
1 str_view(strings, "r{3}")
```

[5] | g<rrr>eat



r appearing exactly 3 times

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select “apple”, “raspberry”, or “blackberry”?  
or “grrreat”?

# More regular expressions

```
1 strings <- c("apple", "banana", "raspberry",  
2             "blackberry", "grrreat", "random")
```

How would I select “apple”, “raspberry”, or “blackberry”?

```
1 str_view(strings, "(.)\\1{1}")
```

```
[1] | a<pp>le  
[3] | raspbe<rr>y  
[4] | blackbe<rr>y  
[5] | g<rr>reat
```

capture  
group

back reference

match the capture  
group (any character)  
repeated once

one letter:  $(.)\backslash 1\{1\}$

## More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "grrreat", "random")
```

How would I select “papa”, “banana”, and “memento”?

$(..)\backslash 1\{1\}$

# More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "grrreat", "random")
```

How would I select “papa”, “banana”, and “memento”?

```
1 str_view(strings, "(..)\1{1}")
```

```
[1] | <papa>  
[2] | b<anan>a  
[3] | <meme>nto
```

```
1 str_view(strings, "(..)+")
```

```
[1] | <papa>  
[2] | <banana>  
[3] | <mement>o  
[4] | <blackberry>  
[5] | <grrrea>t  
[6] | <random>
```



# More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "toboggan", "random")
```

How would I select “banana” and “blackberry”?

# More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "toboggan", "random")
```

How would I select “banana” and “blackberry”?

```
1 str_view(strings, "^b")
```

```
[2] | <b>anana
```

```
[4] | <b>lackberry
```

↑  
anchor

^ means “starts with”

# More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "toboggan", "random")
```

How would I select “papa” and “banana”?

# More regular expressions

```
1 strings <- c("papa", "banana", "memento",  
2             "blackberry", "toboggan", "random")
```

How would I select “papa” and “banana”?

```
1 str_view(strings, "a$")
```

```
[1] | pap<a>  
[2] | banan<a>
```

↑  
anchor  
(ends with)

=> \$ is a metacharacter (not just literal \$)

if we want to match a literal \$  
we need escape characters : \ \$  
in R: \\ \$

# More regular expressions

```
1 "The mean  $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ "
```

How would I extract  $\mu$  and  $\mu = \frac{1}{n} \sum_i x_i$ ?

pattern:      in between      \$

pattern:       $\backslash\$ \underbrace{.+} \backslash\$$   
                 something

# More regular expressions

```
1 "The mean  $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ "
```

How would I extract  $\mu$  and  $\mu = \frac{1}{n} \sum_i x_i$ ?

```
1 str_extract("The mean  $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ "  
2               "\\$.+\\$")
```

```
[1] " $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ "
```

$\underbrace{\hspace{10em}}_{\text{not } \$}$

# More regular expressions

```
1 "The mean  $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ "
```

How would I extract  $\mu$  and  $\mu = \frac{1}{n} \sum_i x_i$ ?

```
1 str_extract_all("The mean  $\mu$  is defined by  $\mu = \frac{1}{n} \sum_i x_i$ ",  
2 "\\$[^\\$]+\\$")
```

```
[[1]]
```

```
[1] " $\mu$ "  
x_i$"
```

Start w/  
\$

↑

↑

↑

end w/  
\$

everything  
except \$

" $\mu = \frac{1}{n} \sum_i$

[ ] character

[^ ] everything except specified  
characters

# More regular expressions

```
1 "The current date (today) is November 3 [2007]."
```

How would I extract “(today)” and “[2007]”?



# More regular expressions

```
1 "The current date (today) is November 3 [2007]."
```

How would I extract "(today)" and "[2007]"?

```
1 str_extract_all("The current date (today) is November 3 [2007].",  
2 "[\\((\\[\\]^\\)\\]+[\\]\\]\\)")
```

[[1]]

[1] "(today)" "[2007]"

Handwritten annotations for the regex pattern:  
- A wavy line under the entire pattern: "[\\((\\[\\]^\\)\\]+[\\]\\]\\)" with the label "pattern".  
- An arrow points to the opening bracket of the first group: "( or [".  
- An arrow points to the closing bracket of the first group: "not ) or ]".  
- An arrow points to the opening bracket of the second group: "end w/ ) or ]".

What if I just want "today" and "2007"?

Handwritten note: "( or [ ..... ) or ]"

^

at the beginning of a pattern  
means "starts with"

anything except  
a ) or ]

[ ^ ] ← everything except specified characters

[ ] ← character class  
e.g. [a-z] → all lowercase letters  
[\\(\\[ ] → either ( or [

# More regular expressions

```
1 "The current date (today) is November 3 [2007]."
```

```
1 str_extract_all("The current date (today) is November 3 [2007].",  
2                  "(?<=[\\(\\[\\^\\)]\\]\\+?(?=[\\)\\]\\])")
```

```
[[1]]
```

```
[1] "today" "2007"
```

# More regular expressions

```
1 "The current date (today) is November 3 [2007]."
```

What if I only want the words?

```
1 str_extract_all("The current date (today) is November 3 [2007].",  
2                 "\\w+")
```

```
[[1]]
```

```
[1] "The"      "current"  "date"     "today"    "is"       "November"
```

```
"3"
```

```
[8] "2007"
```

 ← alphanumeric characters, and \_

# More regular expressions

```
1 "The current date (today) is November 3 [2007]."
```

What if I only want the words?

```
1 str_replace_all("The current date (today) is November 3 [2007].",  
2                  "[^\\w\\s]", "")
```

```
[1] "The current date today is November 3 2007"
```

# Recap: regular expressions

Last time, we learned the following regular expression tools:

- `\d` matches any digit (in R, have to type `\\d` because we write the regex in a string)
- `.` matches any character (except `\n`)
- `+` means “at least once”
- `(?<=)` and `(?=)` are positive lookbehinds and lookaheads
- `|` is alternation (one pattern or another)

# A list of some other useful tools

- `*` means “appears 0 or more times”
- `{m}` means “appears *m* times”
- `\b` is a word boundary (use `\\b` in R)
- `\w` is any alphanumeric character, or underscore (use `\\w` in R)
- `( )` is a capture group
- `[ ]` is a set of characters
- `\s` denotes spaces (use `\\s` in R)
- `^` anchors at the beginning, `$` anchors at the end

