

# Advanced Regular Expressions: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2020

## Syntax

- Loading stringr package:

```
library(stringr)
```

---

## CAPTURE GROUPS

- Extracting text using a capture group:

```
str_match(strings, pattern_with_capture_group)
```

- Extracting text using multiple capture groups:

```
str_match(strings, pattern_with_multiple_capture_groups)
```

- Cleaning text using

```
str_to_lower(strings)  
str_to_upper(strings)
```

---

## SUBSTITUTION

- Substituting a regex match:

```
str_replace(strings, pattern, replacement_text)  
str_replace_all(strings, pattern, replacement_text)
```

---

## CLEANING DATAFRAME

- Using dplyr package:

```
library(dplyr)
```

- Filtering out undesired rows by nesting `filter()` and `str_detect()` functions:

```
filter(str_detect(column_name, pattern))
```

- Creating new column in the dataframe containing extrated mentions of a pattern by nesting `mutate()` and `str_match()` functions:

```
mutate(new_column_name = str_match(column_name, pattern_with_capture_group))
```

- Creating new column in the dataframe containing clean text by nesting `mutate()` and `str_to_lower()` functions:

```
mutate(new_column_name = str_to_lower(column_name))
```

- Selecting and summarizing data chaining `select()`, `group_by()` and `summarise()` functions:

```
new_data <- data %>%
  select(one_column_name, another_column_name, ...) %>%
  group_by(a_column_name_from_the_selection) %>%
  summarise(new_column_name = mean(a_column_name_from_the_selection))
```

## Concepts

- Capture groups allow us to specify one or more groups within our match that we can access separately.

Pattern	Explanation
<code>(yes)no</code>	Matches <code>yesno</code> , capturing <code>yes</code> in a single capture group.
<code>(yes)(no)</code>	Matches <code>yesno</code> , capturing <code>yes</code> and <code>no</code> in two capture groups.

- Backreferences allow us to repeat a capture group within our regex pattern by referring to them with an integer in the order they are captured.

Pattern	Explanation
<code>(yes)no\1</code>	Matches <code>yesnoyes</code>
<code>(yes)(no)\2\1</code>	Matches <code>yesnonoyes</code>

- Lookarounds let us define a positive or negative match before or after our string.

Pattern	Explanation
<code>zzz(?:=abc)</code>	Matches <code>zzz</code> only when it is followed by <code>abc</code>
<code>zzz(?:!abc)</code>	Matches <code>zzz</code> only when it is not followed by <code>abc</code>
<code>(?&lt;=abc)zzz</code>	Matches <code>zzz</code> only when it is preceded by <code>abc</code>
<code>(?!zzz)abc</code>	Matches <code>zzz</code> only when it is not preceded by <code>abc</code>

## Resources

- [R official regex doc](#)
- [List of special characters and classes](#)
- [Overview of regex functions in stringr package](#)
- [Building regular expressions](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2020