# Regular Expression Basics: Takeaways



## Syntax

### REGULAR EXPRESSION IN R

- Loading stringr package:

```
library(stringr)
```

- Searching a string for a regex pattern:

```
str_detect("Rhythm and blues", "blue")
```

- Searching a vector (of strings) for a regex pattern:

```
str_detect(c("Rhythm and blues", "Red light"), "blue")
```

- Counting the mentions of a regex pattern:

```
sum( str_detect( c("Rhythm and blues", "Red light"), "blue") )
```

- Returning the matches pattern from a vector or dataframe:

```
data[str_detect(strings, pattern)]
```

- Extracting the matching string to the pattern:

```
str_extract(strings, pattern)
```

- Extracting a regex capture group from vector:

```
str_match(strings, pattern_with_capture_group)[,2]
```

### USING REGULAR EXPRESSION CLASSES IN R

- double the backslashe to avoid R interpretation error. For example, use `\\w` instead of `\w` .

### ESCAPING CHARACTERS

- Treating special characters as ordinary text using backslashes:

```
\\[pdf\\]
```

## Concepts

- Regular expressions, often referred to as regex, are a set of syntax components used for matching sequences of characters in strings.
- A pattern is described as a regular expression that we write. We say regular expression has matched if it finds the pattern exists in the string.
- Character classes allow us to match certain classes of characters.
- A set contains two or more characters that can match in a single character's position.
- Quantifiers specify how many of the previous characters the pattern requires.
- Capture groups allow us to specify one or more groups within our match that we can access separately.
- Negative character classes are character classes that match every character except a character class.
- An anchor matches something that isn't a character, as opposed to character classes which match specific characters.
- A word boundary matches the space between a word character and a non-word character, or a word character and the start/end of a string.
- Common character classes:

| Character Class | Pattern | Explanation |
|---|---|---|
| Set | `[fud]` | Either `f`, `u`, or `d` |
| Range | `[a-e]` | Any of the characters `a`, `b`, `c`, `d`, or `e` |
| Range | `[0-3]` | Any of the characters `0`, `1`, `2`, or `3` |
| Range | `[A-Z]` | Any uppercase letter |
| Set + Range | `[A-Za-z]` | Any uppercase or lowercase character |
| Digit | `\d` | Any digit character (equivalent to `[0-9]`) |
| Word | `\w` | Any digit, uppercase, or lowercase character (equivalent to `[A-Za-z0-9]`) |
| Whitespace | `\s` | Any space, tab or linebreak character |

- Common quantifiers:

| Quantifier | Pattern | Explanation |
| --- | --- | --- |
| Dot | `.` | Any character except newline |
| Zero or more | `a*` | The character `a` zero or more times |
| One or more | `a+` | The character `a` one or more times |
| Optional | `a?` | The character `a` zero or one times |
| Numeric | `a{3}` | The character a three times |
| Numeric | `a{3,5}` | The character `a` three, four, or five times |
| Numeric | `a{,3}` | The character `a` one, two, or three times |
| Numeric | `a{8,}` | The character `a` eight or more times |

- Common negative character classes:

| Character Class | Pattern | Explanation |
| --- | --- | --- |
| Negative Set | `[^fud]` | Any character except `f`, `u`, or `d` |
| Negative Set | `[^1-3Z\s]` | Any characters except `1`, `2`, `3`, `Z`, or whitespace characters |
| Negative Digit | `\D` | Any character except digit characters |
| Negative Word | `\W` | Any character except word characters |
| Negative Whitespace | `\S` | Any character except whitespace characters |

- Common anchors:

| Anchor | Pattern | Explanation |
| --- | --- | --- |
| Beginning | `^abc` | Matches `abc` only at the start of a string |
| End | `abc$` | Matches `abc` only at the end of a string |
| Word boundary | `s\b` | Matches `s` only when it's followed by a word boundary |
| Word boundary | `s\B` | Matches `s` only when it's not followed by a word boundary |

- Common flags:

| Flag | Pattern | Explanation |
| --- | --- | --- |
| Ignorcase | `(?i)abc` | Matches all different capitalizations of the word `abc` : Abc, ABC, abC, etc |
| Ignoring white spaces and comments | `(?x)a b c` | Matches `abc` |

## Resources

- R official regex doc
- The package `stringr`
- List of special characters and classes
- Overview of regex functions in stringr package
- Building regular expressions