Welcome

INTERMEDIATE REGULAR EXPRESSIONS IN R

Angelo Zehr
Data Journalist

datacamp

Where you might have left off

String Manipulation with stringr in R

From Rebus to writing custom expressions

Does `"cat"` start with `"c"` ?

The rebus way:

```
str_detect("cat", pattern = START %R% "c")
```

Regular expression:

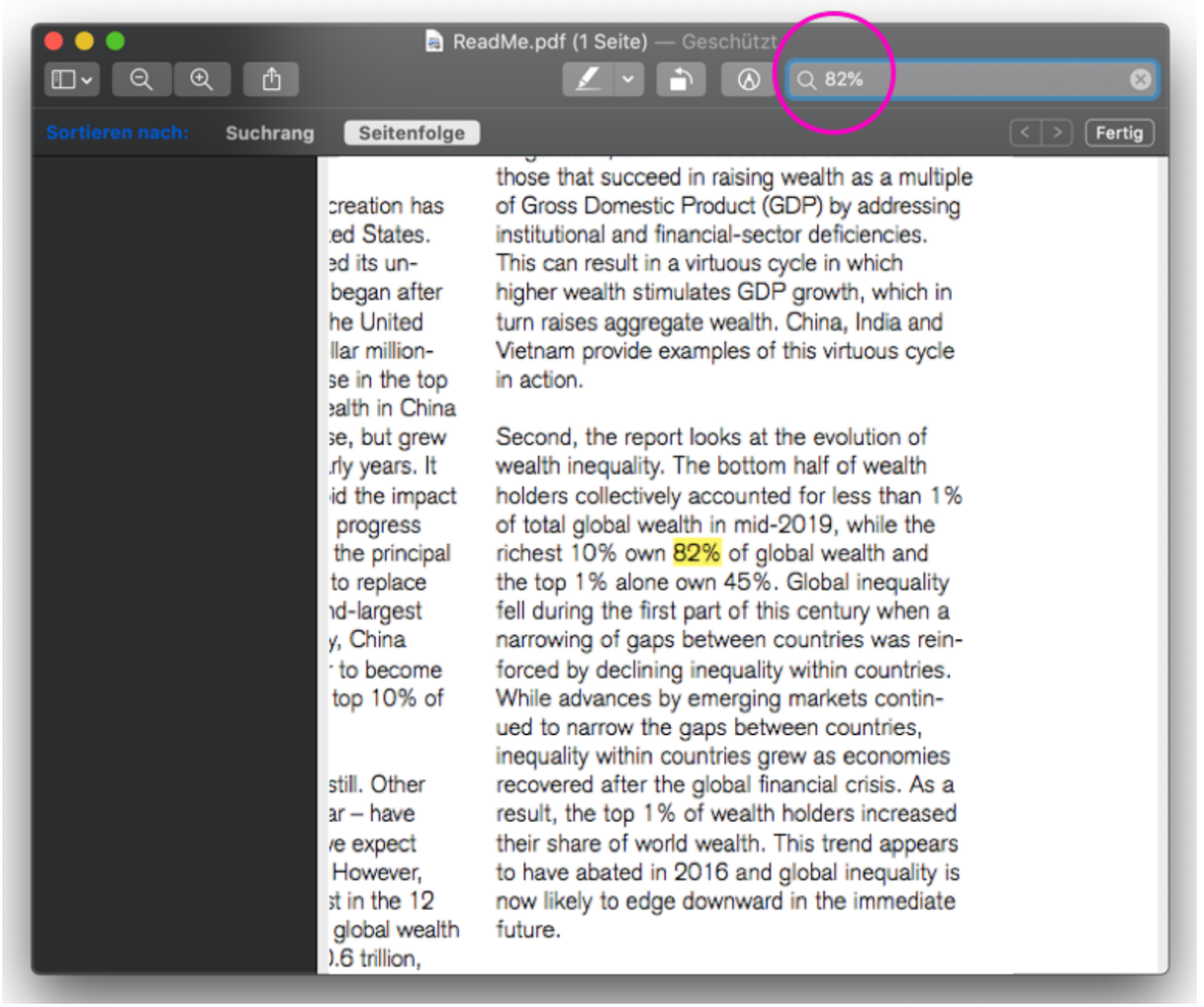
```
str_detect("cat", pattern = "^c")
```

Prerequisites: stringr

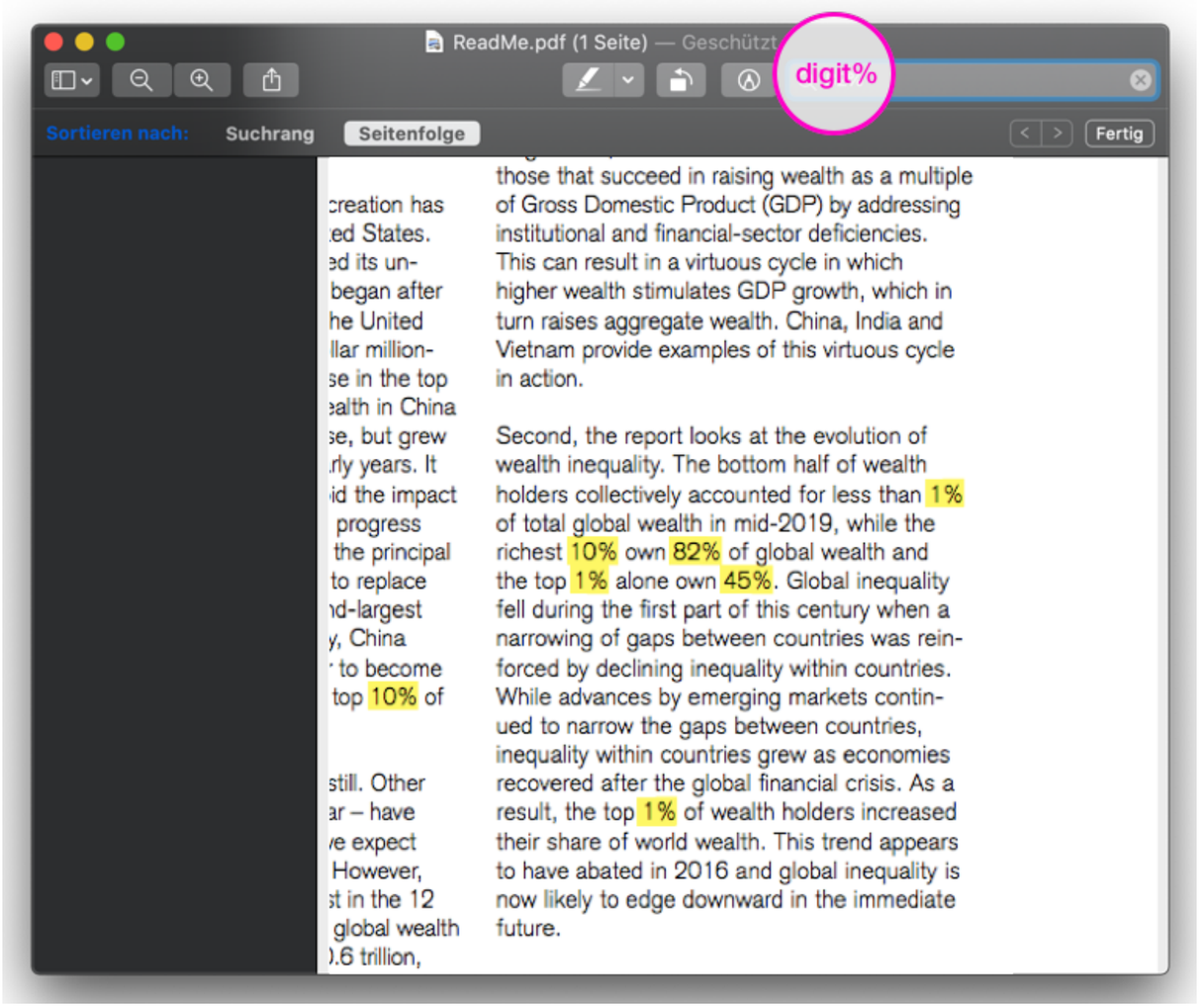
```
str_detect(string, pattern)
```

```
str_match(string, pattern)
```

What regular expressions will help you achieve



What regular expressions will help you achieve



Our first dataset

```
movie_titles <- c(
  "Karate Kid",
  "The Twilight Saga: Eclispe",
  "Knight & Day",
  "Shrek Forever After (3D)",
  "Marmaduke.",
  "Predators",
  "StreetDance (3D)",
  "Robin Hood",
  "Micmacs A Tire-Larigot",
  "Sex And the City 2",
...
```

```
movie_titles[
  str_detect(
    movie_titles,
    pattern = "^K"
  )
]
```

```
"Karate Kid",
"Knight & Day",
...
```

Special characters in regular expressions

Special character	Meaning
`^`	**Caret**: Marks the beginning of a line or string
`$`	**Dollar Sign**: Marks the end of a line or string
`.`	**Period**: Matches anything: letters, numbers or white spaces
`\\.`	**Two backslashes**: Escapes the period when we search an actual period

For example

Code	Result
`str_match("Book", "^.")`	Will match `"B"`
`str_match("Book", ".$")`	Will match `"k"`
`str_match("Book", "\\.")`	No match
`str_match("Book.", "\\.")`	Will match `"."`

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R

Character classes and repetitions

INTERMEDIATE REGULAR EXPRESSIONS IN R

Angelo Zehr
Data Journalist

datacamp

Available character classes

Character Class	Example
`\\d or [:digit:]`	`0, 1, 2, 3,…`
`\\w or [:word:]`	`a, b, c…, 1, 2, 3…, _`
`[A-Za-z] or [:alpha:]`	`A, B, C,…, a, b, c,…`
`[aeiou]`	either `a` , `e` , `i` , `o` or `u`
`\\s or [:space:]`	`" "` , tabs or line breaks

A concrete example

str_match_all()	Result
`"Hi John_35", "\\d"`	`"3", "5"`
`"Hi John_35", "\\w"`	`"H", "i", "J", "o", "h", "n", "_", "3", "5"`
`"Hi John_35", "[A-Za-z]"`	`"H", "i", "J", "o", "h", "n"`
`"Hi John_35", "[aeiou]"`	`"i", "o"`
`"Hi John_35", "\\s"`	`" "`

Repetitions

Syntax	Meaning
`\\w{2}`	exactly 2 times
`\\w{2,3}`	minimum 2 times, maximum 3 times
`\\w{2,}`	minimum 2 times, but no maximum
`\\w+`	1 or more repetitions
`\\w*`	0, 1 or more repetitions

Inversion of character classes

Original	Negation
`\\d` match digits	`\\D` match **all but** digits
`\\w` match word characters	`\\W` match **all but** word characters
`\\s` match spaces	`\\S` match **all but** spaces
`[a-zA-Z]` match alphabet	`[^a-zA-Z]` match **all but** alphabet

Custom pattern with classes

```
str_match_all("Toy Story 3", "[\\d\\s]")
```

Result:

```
     [,1]
[1,] " "
[2,] " "
[3,] "3"
```

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R

The pipe and the question mark

INTERMEDIATE REGULAR EXPRESSIONS IN R

Angelo Zehr
Data Journalist

datacamp

This or that

```
lines <- c(
  "Karate Kid 2, Distributor: Columbia, 58 Screens",
  "Finding Nemo, Distributors: Pixar and Disney, 10 Screens",
  "Finding Harmony, Distributor: Unknown, 1 Screen",
  "Finding Dory, Distributors: Pixar and Disney, 8 Screens"
)
```

```
str_detect(lines, "Columbia|Pixar")
```

```
TRUE TRUE FALSE TRUE
```

Making things optional

```
str_view(lines, pattern = "Distributor|Distributors")
```

```
str_view(lines, pattern = "Distributors?")
```

Karate Kid 2, Distributor: Columbia, 58 Screens

Finding Nemo, Distributors: Pixar and Disney, 10 Screens

Finding Harmony, Distributor: Unknown, 1 Screen

Finding Dory, Distributors: Pixar and Disney, 8 Screens

Greedy vs. lazy

```
str_view("Toy Story 3 In Disney Digital 3D", ".*3")
```

`.*3`


```
str_view("Toy Story 3 In Disney Digital 3D", ".*?3")
```

`.*?3`

Let's practice!

INTERMEDIATE REGULAR EXPRESSIONS IN R