STRING MANIPULATION WITH STRINGR

Capturing

Capturing

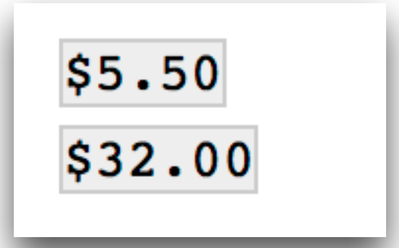
```
> ANY_CHAR %R% "a"
<regex> .a

> capture(ANY_CHAR) %R% "a"
<regex> (.)a

> str_extract(c("Fat", "cat"),
    pattern = ANY_CHAR %R% "a")
[1] "Fa" "ca"

> str_extract(c("Fat", "cat"),
    pattern = capture(ANY_CHAR) %R% "a")
[1] "Fa" "ca"
```

str_match()

```
> str_match(c("Fat", "cat"),
    pattern = capture(ANY_CHAR) %R% "a")
     [,1] [,2]
[1,] "Fa" "F"
[2,] "ca" "c"
```

str_match()

```
> pattern <- DOLLAR %R%
             DGT %R% optional(DGT) %R%
             DOT %R%
             dgt(2)

> str_view(c("$5.50", "$32.00"), pattern = pattern)
```



str_match()

```
> pattern <- DOLLAR %R%
             capture(DGT %R% optional(DGT)) %R%
             DOT %R%
             capture(dgt(2))

> str_match(c("$5.50", "$32.00"), pattern = pattern)
     [,1]     [,2] [,3]
[1,] "$5.50"  "5"  "50"
[2,] "$32.00" "32" "00"
```

Non-capturing groups

```
> or("dog", "cat")
<regex> (?:dog|cat)
```

`dog|cat` **Need parentheses to distinguish** `(dog|cat)` `do(g|c)at`

```
> or("dog", "cat", capture = TRUE)
<regex> (dog|cat)

> capture(or("dog", "cat"))
<regex> ((?:dog|cat))
```

STRING MANIPULATION WITH STRINGR

Let’s practice!

STRING MANIPULATION WITH STRINGR

Backreferences

Backreferences

```
> REF1
<regex> \1

> REF2
<regex> \2
```

In a pattern

```
SPC %R%
one_or_more(WRD) %R%
SPC
```

In a pattern

```
SPC %R%
capture(one_or_more(WRD)) %R%
SPC
```

In a pattern

```
SPC %R%
capture(one_or_more(WRD)) %R%
SPC %R%
REF1
```

```
> str_view("Paris in the the spring",
    SPC %R%
    capture(one_or_more(WRD)) %R%
    SPC %R%
    REF1)
```

```
Paris in the the spring
```

In a replacement

```
> str_replace("Paris in the the spring",
    pattern  = SPC %R%
               capture(one_or_more(WRD)) %R%
               SPC %R%
               REF1,
    replacement = str_c(" ", REF1))
[1] "Paris in the spring"
```

STRING MANIPULATION WITH STRINGR

Let’s practice!

STRING MANIPULATION WITH STRINGR

Unicode and pattern matching

Unicode

- Associates each character with a code point

Character	Code Point
a	61
μ	3BC
😀	1F600

Unicode in R

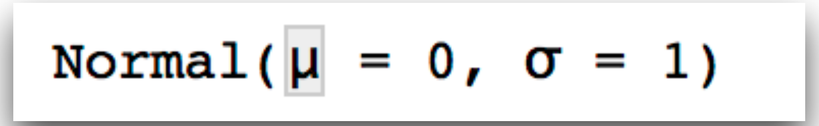
```
> "\u03BC"
[1] "μ"


> "\U03BC"
[1] "μ"

> writeLines("\U0001F44F")
👏
```

Unicode in R

```
> as.hexmode(utf8ToInt("a"))
[1] "61"

> as.hexmode(utf8ToInt("μ"))
[1] "3bc"

> as.hexmode(utf8ToInt("😀"))
[1] "1f600"
```

Matching Unicode

```
> x <- "Normal(\u03BC = 0, \u03C3 = 1)"
> x
[1] "Normal(μ = 0, σ = 1)"

> str_view(x, pattern = "\u03BC")
```



http://unicode.org/charts

http://www.fileformat.info/info/unicode/char/search.htm

Matching Unicode groups

Use \p followed by {name}

Regular expression

```
> str_view_all(x, greek_and_coptic())
```

rebus

```
Normal(μ = 0, σ = 1)
```

```
?Unicode

?unicode_property

?unicode_general_category
```

STRING MANIPULATION WITH STRINGR

Let’s practice!