Advanced String Manipulation

Fall 2022

October 10 2022

Let's start with some positivity ...

```
str_to_lower("BEAUTY is in the EYE of the BEHOLDER")
[1] "beauty is in the eye of the beholder"
```

```
str_to_upper("one small step for man, one giant leap for mankind")
[1] "ONE SMALL STEP FOR MAN, ONE GIANT LEAP FOR MANKIND"
```

```
str_to_title("Aspire to inspire before we expire")
[1] "Aspire To Inspire Before We Expire"
```

```
str_to_sentence("everything you can imagine is real")
[1] "Everything you can imagine is real"
```

Some more regexes

```
aboutMe <- c("My phone number is 236-748-4508.")
```

```
str_view_all(aboutMe, "\\.") # literal period "."
```

```
My phone number is 236-748-4508.
```

```
str_view_all(aboutMe, "[^(\\d)(\\s)(\\-)(\\.)]") # everything except
```

```
My phone number is 236-748-4508.
```

Alternates: OR

```r
aboutMe <- c("My phone number is 236-748-4508.")
```

```r
str_view(aboutMe,"8|6-")
```

```
My phone number is 236-748-4508.
```

```r
str_view(aboutMe,"(8|6-)")
```

```
My phone number is 236-748-4508.
```

```r
str_view_all(aboutMe,"(8|6)-")
```

```
My phone number is 236-748-4508.
```

More Duplicating Groups

```r
foo <- c("addidas", "missim")
```

```r
# anything then repeat anything
str_view(foo, "(.)\\1")
```

```
addidas

missim
```

```r
# strings like `xyzzyx`
str_view(foo, "(.)(.)(.)\\3\\2\\1")
```

```
addidas

missim
```

```r
str_view(foo, "(.)(.)\\1")
```

```
addidas

missim
```

Finding patterns

```
# find the last word in a sentence
str_view_all("it's a goat.",
             "[a-z]+\\.")
```

```
it's a goat.
```

```
# find word with  `'s`
str_view_all("it's a goat.",
             "[a-z]+\\'\\w")
```

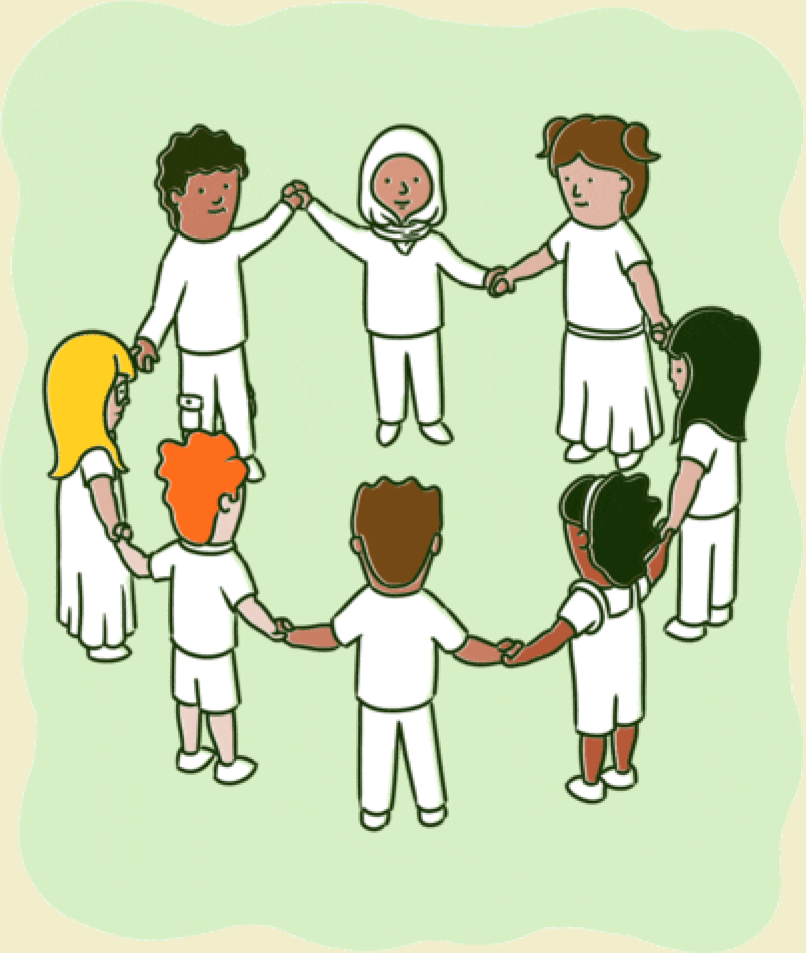
```
it's a goat.
```

```
# find a single letter word separated by spaces
str_view_all("it's a goat.",
             "(\\s)(\\w)\\s")
```

```
it's a goat.
```

Group Activity 1

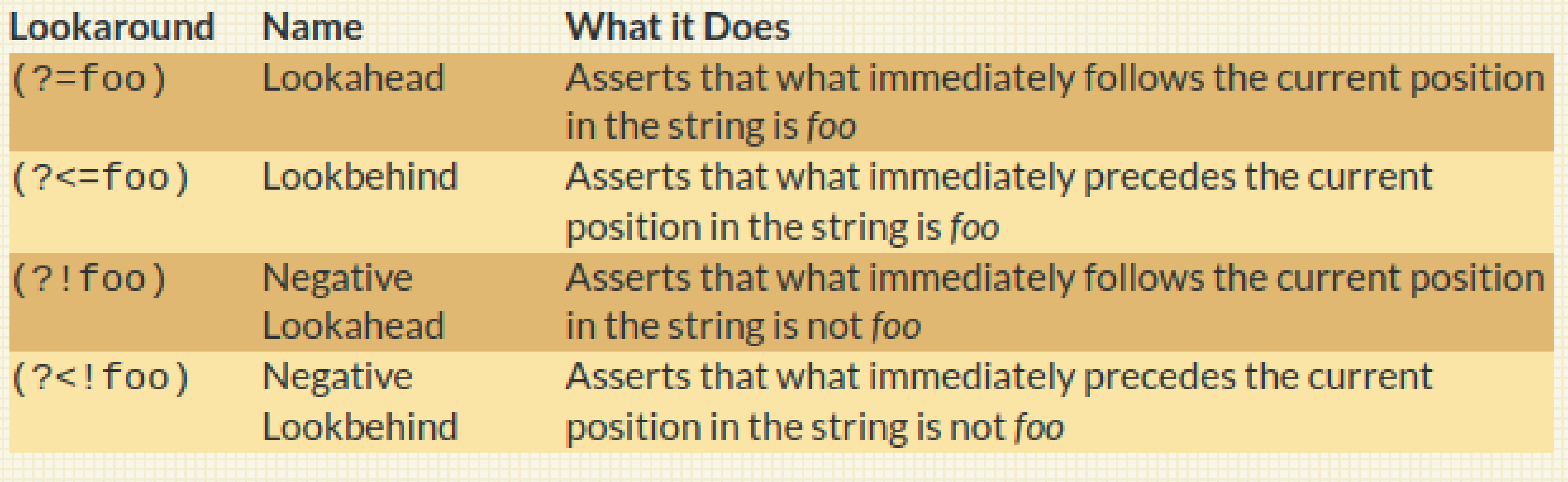
06:00



- Let's go over to maize server/ local Rstudio and our class moodle
- Get the class activity 13.Rmd file
- Work on activity 1
- Knit to .html as .pdf won't work

Look ahead and look behind !!

What are these?



Lookaround	Name	What it Does
`(?=foo)`	Lookahead	Asserts that what immediately follows the current position in the string is *foo*
`(?<=foo)`	Lookbehind	Asserts that what immediately precedes the current position in the string is *foo*
`(?!foo)`	Negative Lookahead	Asserts that what immediately follows the current position in the string is not *foo*
`(?<!foo)`	Negative Lookbehind	Asserts that what immediately precedes the current position in the string is not *foo*

Source: click here

Look ahead example

Positive look ahead operator `x(?=[y])` will find `x` when it comes before `y`

Negative version is `x(?![y])` (`x` when it comes before something that isn’t `y`)

```r
str_view_all("it's a goat.", "t(?=[\\.])") # t before a period
```

```
it's a goat.
```

Look ahead example

Positive look ahead operator `x(?=[y])` will find `x` when it comes before `y`

Negative version is `x(?![y])` (`x` when it comes before something that isn’t `y`)

```r
str_view_all("it's a goat.","[a-z]+(?=[\\.])") # 1+ letters before a period
```

```
it's a goat.
```

Look behind example

Positive look behind operator `(?<=[x])y` will find `y` when it follows `x`

Negative version is `(?<![x])y` (`y` when it does not follow `x`)

```r
str_view_all("that is a top cat.","(?<=[a-z])t+") # one or more t, if preceded by a letter
```

```
that is a top cat.
```

Look behind example

Positive look behind operator `(?<=[x])y` will find `y` when it follows `x`

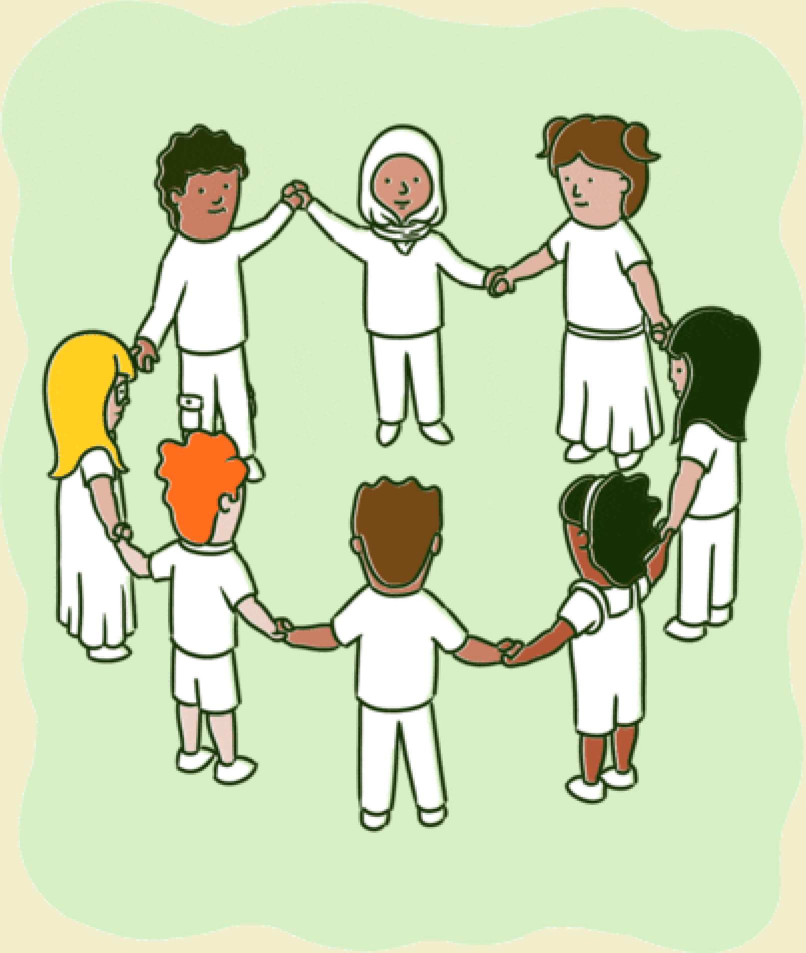
Negative version is `(?<![x])y` (`y` when it does not follow `x`)

```r
# t and one or more letter not preceded by a letter
str_view_all("that is a top cat.","(?<![a-z])t[a-z]+")
```

```
that is a top cat.
```

Group Activity 2

08:00



- Go back to the activity file
- Continue working on activity 3
- Ask me questions

Analyzing Trump tweets

What proportion of tweets (text) mention “Hillary” or “Clinton”?

```
tweets %>%
  summarize(prop = mean(str_detect(str_to_lower(text),"hillary|clinton")))
# A tibble: 1 × 1
   prop
  <dbl>
1 0.174
```

- About 17.4% of these tweets mention Hillary or Clinton.

How are the hashtags used?

```r
tweets %>%
  mutate(ct = str_count(text, "#")) %>%
  select(ct, text) %>%
  summarize(prop = mean(ct > 0))
```

```
# A tibble: 1 × 1
   prop
  <dbl>
1 0.283
```

Finding URLs

URLs in tweets start with https://t.co/ followed by a string of letters or numbers

```
link <- "https://t.co/[A-Za-z\\d]+"
tweets$text[992]
[1] "I LOVE NEW YORK! #NewYorkValues \r\nhttps://t.co/dbTDhYAX1v"
```

```
str_view(tweets$text[992], link)
```

```
I LOVE NEW YORK! #NewYorkValues https://t.co/dbTDhYAX1v
```

What proportion of tweets have links?

```
tweets %>%
  summarize(prop = mean(str_detect(text, link)))
# A tibble: 1 × 1
   prop
  <dbl>
1 0.342
```

- about 34.2% of tweets have a link.

Removing links from tweets

```
tw_noLink <- tweets %>%
  mutate(textNoLink = str_replace_all(text, link, ""))
```

```
tw_noLink$text[992]
[1] "I LOVE NEW YORK! #NewYorkValues \r\nhttps://t.co/dbTDhYAX1v"
tw_noLink$textNoLink[992]
[1] "I LOVE NEW YORK! #NewYorkValues \r\n"
```

Get the tweets with links

```
tweets %>%
  filter(str_detect(text, link)) %>%
  select(text)
# A tibble: 517 × 1
   text
   <chr>
 1 "Join me in Fayetteville, North Carolina tomorrow evening at 6pm. Tickets no…
 2 "#ICYMI: \"Will Media Apologize to Trump?\" https://t.co/ia7rKBmioA"
 3 "Thank you Windham, New Hampshire! #TrumpPence16 #MAGA https://t.co/ZL4Q01Q4…
 4 ".@Larry_Kudlow - 'Donald Trump Is the middle-class growth candidate'\r\nhtt…
 5 "#CrookedHillary is not fit to be our next president! #TrumpPence16 \r\nhttp…
 6 "Good luck #TeamUSA\r\n#OpeningCeremony #Rio2016 https://t.co/mS8qsQpJPh"
 7 "'Trump is right about violent crime: It\x92s on the rise in major cities'\r…
 8 "Thank you Green Bay, Wisconsin! Governor @Mike_Pence and I will be back soo…
 9 "DON'T LET HILLARY CLINTON DO IT AGAIN!\r\n#TrumpPence16\r\nhttps://t.co/1mG…
10 "Thank you Des Moines, Iowa! Governor @Mike_Pence and I appreciate your supp…
# … with 507 more rows
```

Extract all tweets with links

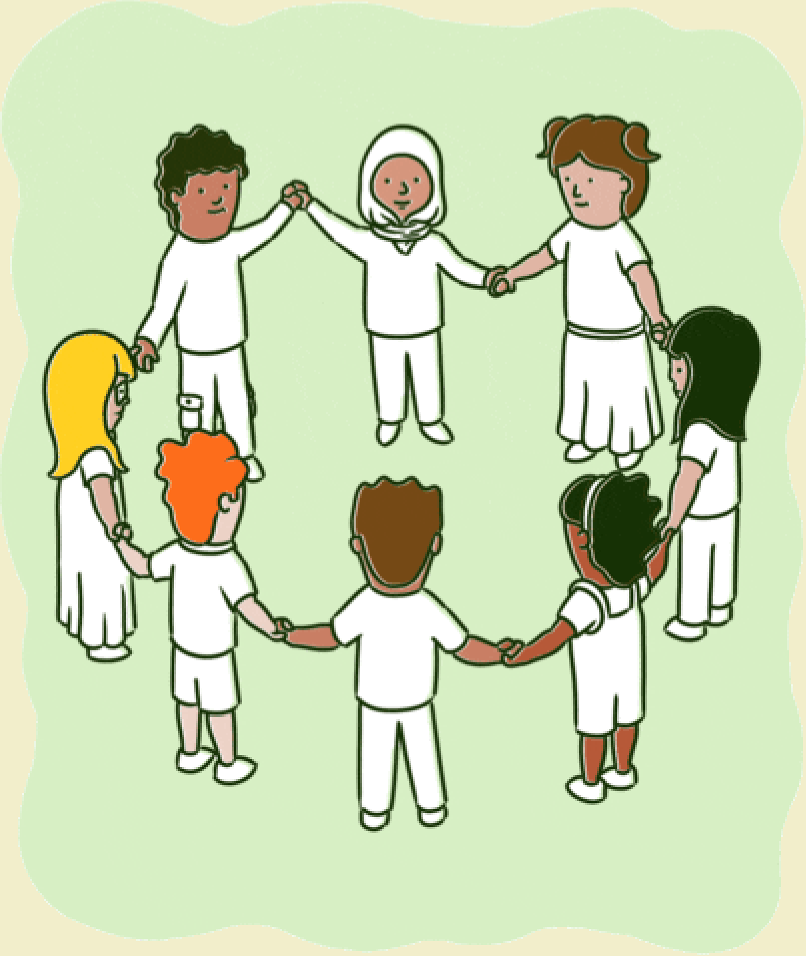
```
tweets %>% select(text) %>%
  str_extract_all(link)
[[1]]
  [1] "https://t.co/Z80d4MYIg8" "https://t.co/ia7rKBmioA"
  [3] "https://t.co/ZL4Q01Q49s" "https://t.co/YbqkhWNm0g"
  [5] "https://t.co/I0zJO2sZKk" "https://t.co/mS8qsQpJPh"
  [7] "https://t.co/XbnZ5vktGk" "https://t.co/qsYbyrm3UR"
  [9] "https://t.co/1mGkPNZPKF" "https://t.co/gr6tGqqmcm"
 [11] "https://t.co/5yuLKyh8Q6" "https://t.co/3EzG620fpT"
 [13] "https://t.co/jsAMGO3s4P" "https://t.co/3Hcnzj0Slx"
 [15] "https://t.co/sEwLWkn1Sz" "https://t.co/UODSMp0oTo"
 [17] "https://t.co/oVfF28rWL5" "https://t.co/RhblAXkNPw"
 [19] "https://t.co/hr4O8Xgq2R" "https://t.co/Iui1F2z9ca"
 [21] "https://t.co/3Hcnzj0Slx" "https://t.co/sEwLWkn1Sz"
 [23] "https://t.co/0Ei3EdQdXB" "https://t.co/xrTQjt9WOC"
 [25] "https://t.co/VSnBoQYoZs" "https://t.co/Al5bZlRFYk"
 [27] "https://t.co/QoxJf4Xzbc" "https://t.co/IAcLfXe463"
```

Unlist the list entries

```
tweets %>% select(text) %>%
  str_extract_all(link) %>%
  unlist()                    # unlist and coerce into a vector
  [1] "https://t.co/Z80d4MYIg8" "https://t.co/ia7rKBmioA"
  [3] "https://t.co/ZL4Q01Q49s" "https://t.co/YbqkhWNm0g"
  [5] "https://t.co/I0zJO2sZKk" "https://t.co/mS8qsQpJPh"
  [7] "https://t.co/XbnZ5vktGk" "https://t.co/qsYbyrm3UR"
  [9] "https://t.co/1mGkPNZPKF" "https://t.co/gr6tGqqmcm"
 [11] "https://t.co/5yuLKyh8Q6" "https://t.co/3EzG620fpT"
 [13] "https://t.co/jsAMGO3s4P" "https://t.co/3Hcnzj0Slx"
 [15] "https://t.co/sEwLWkn1Sz" "https://t.co/UODSMp0oTo"
 [17] "https://t.co/oVfF28rWL5" "https://t.co/RhblAXkNPw"
 [19] "https://t.co/hr4O8Xgq2R" "https://t.co/Iui1F2z9ca"
 [21] "https://t.co/3Hcnzj0Slx" "https://t.co/sEwLWkn1Sz"
 [23] "https://t.co/0Ei3EdQdXB" "https://t.co/xrTQjt9WOC"
 [25] "https://t.co/VSnBoQYoZs" "https://t.co/Al5bZlRFYk"
 [27] "https://t.co/QoxJf4Xzbc" "https://t.co/IAcLfXe463"
 [29] "https://t.co/QvSC5RLFMw" "https://t.co/uVeAzY2zqp"
```

Group Activity 3

08:00



- Continue working on activity 3
- Ask me questions