Regulax Expressions Lab in `stringr`

Paul Intrevado

August 09, 2018

Question 1

Using the `state.name` data, do the following:

a) Find all states that contain at least one instance of he letter "z"

```
str_subset(state.name, "z")
```

```
## [1] "Arizona"
```

b) Find all states that contain two subsequent "s"'s, i.e.,"ss"

```
str_subset(state.name, "ss")
```

```
## [1] "Massachusetts" "Mississippi"   "Missouri"      "Tennessee"
```

c) Find all states that contain at least one but not more than two subsequent "s"'s, i.e.,"s" or "ss"

```
str_subset(state.name, "s{1,2}")
```

```
##  [1] "Alaska"        "Arkansas"      "Illinois"      "Kansas"
##  [5] "Louisiana"     "Massachusetts" "Minnesota"     "Mississippi"
##  [9] "Missouri"      "Nebraska"      "New Hampshire" "New Jersey"
## [13] "Pennsylvania"  "Rhode Island"  "Tennessee"     "Texas"
## [17] "Washington"    "West Virginia" "Wisconsin"
```

d) Find all states that contain at lest two "i"'s, but the they need not be subsequent

```
str_subset(state.name, "i\\w*i")
```

```
##  [1] "California"    "Hawaii"        "Illinois"      "Louisiana"
##  [5] "Michigan"      "Mississippi"   "Missouri"      "Virginia"
##  [9] "West Virginia" "Wisconsin"
```

e) Find all states that have compound names, i.e., have two words separated by a space

```
str_subset(state.name, " ")
```

```
##  [1] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
##  [5] "North Carolina" "North Dakota"   "Rhode Island"   "South Carolina"
##  [9] "South Dakota"   "West Virginia"
```

f) Find all states that begin with either "North" or "South"

```
str_subset(state.name, "(North|South)")
```

```
## [1] "North Carolina" "North Dakota"   "South Carolina" "South Dakota"
```

Question 2

Using `fruit` data from stringr package, do the following:

a) How many fruits are melons?

```
str_subset(fruit, "melon")
```

```
## [1] "canary melon" "rock melon"   "watermelon"
```

b) How many fruits are berries?

```
str_subset(fruit, "berry")
```

```
##  [1] "bilberry"    "blackberry"  "blueberry"   "boysenberry" "cloudberry"
##  [6] "cranberry"   "elderberry"  "goji berry"  "gooseberry"  "huckleberry"
## [11] "mulberry"    "raspberry"   "salal berry" "strawberry"
```

c) How many berries come from a single word versus a compound word?

```
length(str_subset(fruit, " berry"))
```

```
## [1] 2
```

```
length(str_subset(fruit, "berry"))
```

```
## [1] 14
```

Question 3

Locate 1st sequence of 1 or more consecutive numbers in the following character vector:

```
x <- c("abcd", "a22bc1d", "ab3453cd46", "a1bc44d")
```

```
str_locate(x, "\\d+")
```

```
##      start end
## [1,]    NA  NA
## [2,]     2   3
## [3,]     3   6
## [4,]     2   2
```

Question 4

a) Write a regular expression that will match a typical US phone number.

```
"^//d{3}-//d{3}-//d{4}$"
```

b) What if that phone number begins with a "+1"?

```
"^(\\+1)? ?//d{3}-//d{3}-//d{4}$"
```

c) What is the phone number lacks spaces?

```
"^(\\+1)? ?//d{3}-?//d{3}-?//d{4}$"
```

d) What if the area code is wrapped in round braces?

```
"^(\\+1)? ?(\\(?//d{3}\\)|//d{3})-?//d{3}-?//d{4}$"
```

e) What if, instead of dashes, someone decides to use spaces or a "."?

```
"^(\\+1)? ?(\\(?//d{3}\\)|//d{3})[-\\.]?//d{3}[-\\.]?//d{4}$"
```

Question 5

Write a regular expression that will match a gmail address.

```
"^\\w+@gmail.com$"
```