if inside of reactive, remember to add
() ⇒ because it is a f^n in RV

can also see a wizard for random seed.

Lec 20 Regular Expressions (regex)   10/14.

Define a certain "amount of text" : pattern.
Pattern matching

Use Letter, digits, and symbols.

- Literal characters (easy part) : upper lower case letters
  digits, and a few symbol. They match themselves.

- Meta character : set of character (Typically symbol)
  that a special meaning.

All f^n of strings start with 'str_'	txt1 = c("car", "bus", 'bike', ..., 'airplane') str_view (txt1, Pattern = 'a') ⇒ inspect everything in data to find "a" ↓ if Match, highlight the matching

Function from "stringr"

- str_view()

- str_detect()

only change the first match → str_replace() → str_replace(txt1, pattern = 'a', replacement = 'A') → replace the letter

- str_extract()

- str_split()

change all of them → str_replace_all()

txt2 = c("5", "5.00", "5-0", "500", "555", "550")

#A Metacharacter = "." (dot).

Wildcard metacharacter: matches any other character.
str_view(txt2, pattern = ".") → select all elements of txt2.

#A Backslash: escape metacharacter "\"

str_view(txt2, pattern = '\\.') → Need to write "\\" instead of one

character sets

We define a set with square brackets: []
example: the set of lower case vowels.

str_view(txt1, pattern = "[aeiou]")

character ranges =
.'[0-9]' = '[0123456789]'

.'[a-z]' = '[a---z]'

.'[A-Z]' = '[A...Z]'

- alphanumeric range = '[0-9a-fA-F]' hexadecimal